AFRL-IF-WP-TR-2006-1532

MULTI-TIMESCALE COMPLEX ADAPTATION

James Schwaber
Raj Vadigepalli
Praveen Chakravarthula

Thomas Jefferson University
102 Walnut St.
Philadelphia, PA 19107-5587

MARCH 2006

Final Report for 05 September 2001 – 31 December 2005

Approved for public release; distribution is unlimited.

STINFO COPY

INFORMATION DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7334

NOTICE AND SIGNATURE PAGE

AFRL-IF-WP-TR-2006-1532 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//Signature//

JAMES B. MONCRIEF, Proj Eng
Embedded Information Systems Branch
Advanced Computing Division
Information Directorate

//Signature//

JAMES S. WILLIAMSON, Chief
Embedded Information Systems Branch
Advanced Computing Division
Information Directorate

//Signature//

WALTER B. HARTMAN, Actg Chief
Wright Site
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
March 2006	Final	09/05/2001 – 12/31/2005

4. TITLE AND SUBTITLE
MULTI-TIMESCALE COMPLEX ADAPTATION

5a. CONTRACT NUMBER
F30602-01-2-0578

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61101E

6. AUTHOR(S)
James Schwaber
Raj Vadigepalli
Praveen Chakravarthula

5d. PROJECT NUMBER
BIOC

5e. TASK NUMBER
M2

5f. WORK UNIT NUMBER
94

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Thomas Jefferson University
102 Walnut St.
Philadelphia, PA 19107-5587

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Information Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson AFB, OH 45433-7334

Defense Advanced Research Projects
Agency/Information Processing
Technology Office (DARPA/IPTO)
3701 Fairfax Drive
Arlington, VA 22203

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL-IF-WP

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-IF-WP-TR-2006-1532

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Report contains color. PAO Case Number: AFRL/WS 06-2112, 30 Aug. 2006.

14. ABSTRACT

The overall goal of the project was to develop structured approaches to modeling complex gene regulation dynamics underlying cellular adaptation in mammalian systems. This involves integration of two erstwhile disjoint aspects: mathematical modeling and bioinformatics of high-throughput biological data. As part of a structured approach to tackle this problem, we have developed methods and software tools for identification of 1) robust patterns of gene expression using a meta-clustering approach, 2) network structures from these patterns, and 3) a continuous-time regulatory network model based on temporally discrete gene expression data and predicted network structures. A web-based, graphical user interface was developed for the network structure prediction software, PAINT, and has been released as a DARPA BioSPICE module. We have successfully employed our structured approach in the study of various gene regulatory networks from *in silico* model systems, yeast cell cycle, neuronal differentiation and adaptation, circadian rhythms, and cellular response to pathogens.

15. SUBJECT TERMS
bioinformatics, biological modeling, circadian rhythms, gene regulation, meta-clustering, BioSPICE, Promoter Analysis and Interactive Network Toolset (PAINT)

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			James B. Moncrief
Unclassified	Unclassified	Unclassified	SAR	76	19b. TELEPHONE NUMBER *(Include Area Code)* (937) 255-6548 x3606

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

Table of Contents

Figures

Tables

Acknowledgements

This work has been performed under financial support from DARPA BioCOMP program (F30602-01-2-0578), PI: Dr. James S. Schwaber, Thomas Jefferson University. We would like to thank Dr. Sri Kumar and his associates for their support throughout the project duration.

1 Summary

The overall goal of the project is to develop structured approaches to modeling complex gene regulation dynamics underlying cellular adaptation in mammalian systems. This involves integration of two erstwhile disjoint aspects: mathematical modeling and bioinformatics of high-throughput biological data. As part of a structured approach to tackle this problem, we have developed methods and software tools for identification of 1) robust patterns of gene expression using a meta-clustering approach, 2) network structures from these patterns, 3) continuous-time regulatory network model based on temporally discrete gene expression data and predicted network structures. A web-based, graphical user interface was developed for the network structure prediction software, PAINT, and has been released as a DARPA BioSPICE module. We have successfully employed our structured approach in the study of various gene regulatory networks from *in silico* model systems, yeast cell cycle, neuronal differentiation and adaptation, circadian rhythms, and cellular response to pathogens.

2 Introduction

Over the past decade, technological developments have resulted in rapidly growing public resources containing systematic data sets of various types: gene expression changes from microarrays; protein-DNA interaction and transcription factor (TF) activity data from protein binding assays, chromatin immunoprecipitation (ChIP) experiments (Wells and Farnham, 2002), and DNA footprinting (Kang et al., 2002; Ricci and El-Deiry, 2003); protein-protein interactions from two hybrid experiments and coimmunoprecipitation; and, genomic sequence, annotation and ontology information in public databases. The analysis of these large datasets holds the promise of identification of the nonlinear dynamic systems function of the interconnected gene and biochemical regulatory networks.

Attempts at reverse engineering the gene regulatory networks from microarray data alone have met with varied success (Holstege et al., 1998; D'Haeseleer et al., 1999; Wessels et al., 2001; Ronen et al., 2002). Typically, all the genes are considered as potentially regulating all the other genes and the suboptimal and nonunique results are subsequently pruned either by setting thresholds on the quantitative parameters representing interaction strength or via constrained optimization (Yeung et al., 2002). Combining the available heterogeneous data types significantly improves the ability to unravel the regulatory networks (Tavazoie et al., 1999; Hughes et al., 2000; Zak et al., 2001; Hartemink et al., 2002; Ideker et al., 2002). The principal effect of incorporating additional data types apart from microarrays is to constrain the number of regulatory interactions per gene. Based on the known protein-DNA and protein-protein interactions, many interactions can be required to be present or specified to be nonexistent in the identification algorithm. This limits the number of interaction parameters to search for and renders the network identification algorithms tractable for a large number of genes (Zak et al., 2001; Yueng et al., 2002).

The biological mechanism of transcriptional regulation is by specific transcription factors (TFs) binding to the transcriptional regulatory elements (TREs) present in the cis-regulatory region (promoter) of the corresponding genes. The binding is sequence specific and the binding sites are present on multiple genes. This results in an interconnected transcriptional regulatory network. Hence, the analysis of the promoters for the genes of interest for known and predicted TF binding sites will directly provide a good candidate set of network interactions (Hughes et al. 2000; Hartemink et al. 2002; Ideker et al. 2002). This approach has been most successful in developing a detailed understanding of gene regulatory networks in yeast (Tavazoie et al., 1999; Ideker et al., 2001). The availability of genomic sequence combined with extensive information about TF binding site motifs has enabled system-wide analyses to unravel the gene regulatory networks that govern the response of yeast to a multitude of environmental perturbations (Tavozie et al., 1999; Ideker et al., 2001). Similar efforts are in progress in Drosophila (Berman et al., 2002), sea urchin (Davidson et al., 2002) and human systems (Elkon et al., 2003).

In this context, the objective of the bioinformatics research efforts described here is to develop an automated and scalable bioinformatics approach to the identification and analysis of candidate regulatory interactions in a specific experimental setting. We have developed the Promoter Analysis and Interaction Network Tool (PAINT) for Transcriptional Regulatory Network Analysis (TRNA). Briefly, PAINT processes a list of unique identifiers representing the genes of interest and produces an interaction matrix that represents a candidate set of interactions between the transcription factors and the genes. This information can be subsequently employed in various network identification, analysis and visualization software. The objective is not to

develop another tool for sequence analysis, but to construct a modular and extensible platform into which various sequence analysis tools, network analysis and visualization software, and model identification tools can be 'plugged in'. In addition to the command line and the web-based interfaces (available at http://www.dbi.tju.edu/dbi/tools/paint), Early versions of PAINT modules were accessible as the 'agents' that communicate via Open Agent Architecture in the DARPA BioSPICE platform (http://www.biospice.org). Subsequently, PAINT has been available as a Dashboard module in BioSPICE. The details of this module are given in Section X.

On the mathematical modeling end, the broad conceptual postulate that systems engineering techniques developed for complex chemical processes may be applicable to complex cell biological processes is very compelling. However, a naïve, "direct" application of systems engineering techniques to biological problems of practical significance may be rendered virtually ineffective by fundamental differences between cell biology and chemical processes. These differences and the problems they pose are illustrated in a case study below (Section 3.0) on modeling a gene regulatory network involved in the yeast cell cycle. Complete details of the case study are available in Zak et al., (2003). We demonstrate how the biological essence complicates a straightforward "process modeling/identification" problem and subsequently recommend an alternative approach. The approach—a middle ground between a direct, "off the shelf" application of systems engineering tools and a "one-at-a-time" ad-hoc development — incorporates fundamental knowledge of the mechanisms and constraints intrinsic to biological systems**.**

A schematic of the TRNA workflow is shown in the Figure 1. Deriving the regulatory structure from gene expression data is handled by PAINT, while predicting the regulatory activity based on the structure and the observed gene expression can be done using Network Component Analysis (NCA, Liao et al., 2003) or Karyote Genome Analyzer (KAGAN) tools on the BioSPICE Dashboard.

The rest of the document is structured as follows: The problems underlying gene regulatory network modeling and a summary of the structured approach we have developed are presented in Section 3. Next, the details of the robust clustering of gene expression data are presented in Section 4. The details of the bioinformatics tool PAINT for network structure prediction are presented in Section 5. In Sections 4 and 5, the descriptions of the software and associated BioSPICE modules are followed by the details of their application in specific case studies. Section 6 presents a detailed structured approach incorporating all the above methods and demonstrates it in a yeast cell cycle case study where all the relevant system-wide data sets are available in one single biological system.

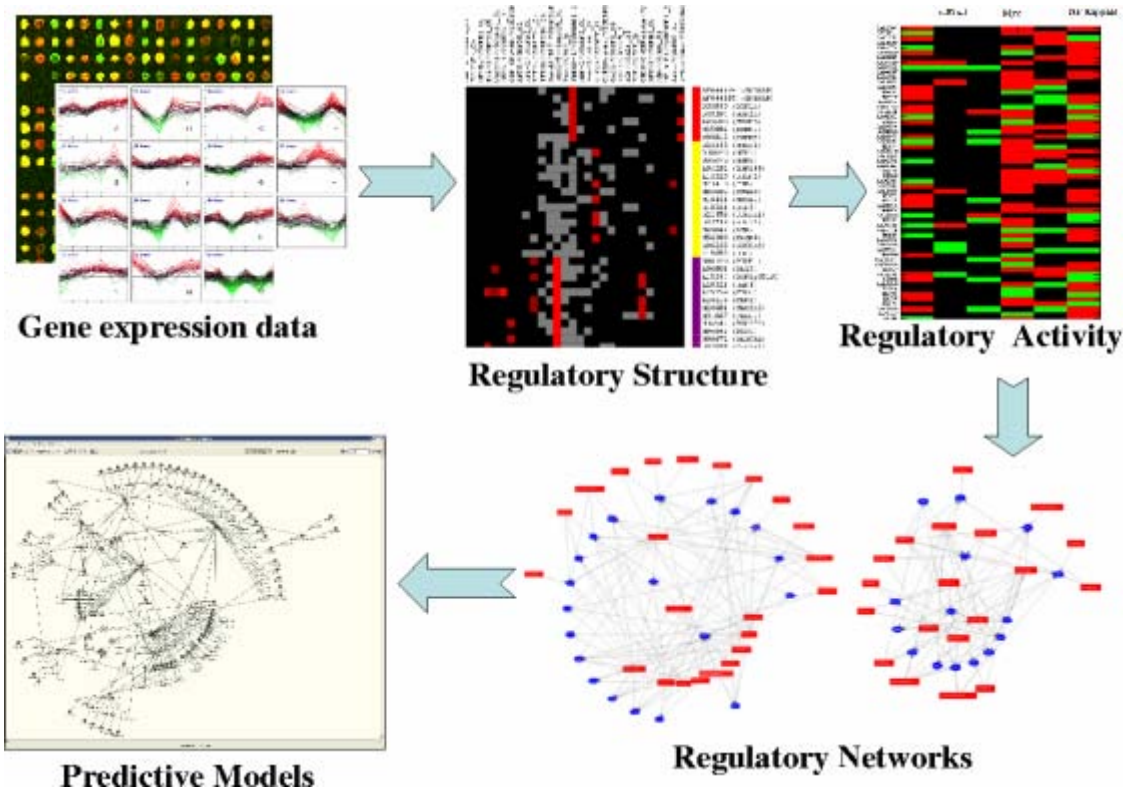

Figure 1: A schematic of the Transcriptional Regulatory Network Analysis workflow.

3 Modeling and Identification of Gene Regulatory Networks

The objective of gene regulatory network identification is to enable the scientist to go beyond merely observing the qualitative changes in gene activities, and actually infer and quantify causal links between the genes that underlie physiological responses. Computational models of these causal links between genes can provide system-wide understanding of the regulation that is fundamental to all life processes, and accelerate efforts in deciphering the structure of complex biological systems. Ultimately, such models may be used for generating testable hypotheses about novel drug targets for prevention and treatment of complex diseases.

Mathematically, the gene network identification problem may be formulated as the identification of the vector function $\mathbf{f}(\cdot)$ and the estimation of the parameter vector $\mathbf{p}$ in Equation 1, given measurements of the expression levels (mRNA levels) over time, $\mathbf{x}(t)$, and the external input perturbations, $\mathbf{v}(t)$, that initiated this observed response:

$$d\mathbf{x}/dt = \mathbf{f}(\mathbf{x},\mathbf{v},\mathbf{p}) \qquad (1)$$

From this formulation, the gene regulatory network identification problem appears to be a straightforward modeling/identification problem. However, given the complexity of the biological processes Equation 1 describes, and given the technical issues associated with the measurement of gene expression, the problem is not so straightforward. First is the issue of scale: the vector of expression levels $\mathbf{x}(t)$ at each sample point in time is of dimension N_g, where N_g is the number of genes in the genome of the organism being studied. For yeast, $N_g \sim 6{,}000$, while for humans, $N_g \sim 25{,}000$, making Equation 1 several orders of magnitude larger in scale than what is typical for chemical process identification problems (for a database of typical system identification problems, see: *http://www.esat.kuleuven.ac.be/sista/daisy* (De Moor et al., 1997)). This issue of scale is further compounded as follows. In standard chemical process identification, the process model is based entirely on $\mathbf{x}(t)$ and $\mathbf{v}(t)$ data. However, for the biological systems considered presently, using only $\mathbf{x}(t)$ and $\mathbf{v}(t)$ data for system identification requires making allowance for all genes to interact with all genes, giving rise to a model with *at least* $N_g \times N_g$ parameters that must be estimated from data. Of course, the fully connected network is not realistic because biological networks are known to be sparsely interconnected (Arnone and Davidson, 1997; Jeong et al., 2001; Ravasz et al., 2002) and thus the majority of the parameters will be zero. The identification of Equation 1 is thus a problem in simultaneous structure and parameter identification on a scale that is extremely rare for chemical process systems. Lastly are the data requirement issues. The problems, associated with the quality and quantity of gene expression data, are extreme compared to what is typical with chemical process data. With the current state of the art, each sample of $\mathbf{x}(t)$ is highly contaminated with noise (Nadon and Shoemaker, 2002; Sebastiani et al., 2003); furthermore, because of the expense and difficulty in acquiring gene expression data, the number of time points N_m at which $\mathbf{x}(t)$ measurements are taken is relatively small ($N_m << N_g$). Thus, from the classical process identification perspective, the gene regulation network identification problem has the following characteristics:

1. It consists of a several thousand ordinary differential equations,
2. with an unknown structure and with potentially millions of parameters,
3. for which only a limited amount of relatively poor quality data is available.

These issues of scale, simultaneous structure and parameter identification, and limited data quality and quantity make the gene regulatory network identification problem highly unconventional compared to the equivalent chemical process problem.

3.1 Unstructured Approaches

In spite of the challenges outlined above, numerous approaches to the gene regulatory network identification problem have been reported in the literature, with the vast majority formulated as described above, depending only on $\mathbf{x}(t)$ and $\mathbf{v}(t)$ data (D'Haeseleer et al., 2000; Brazhnik et al., 2002). For reasons that will become clear later, we call these approaches *unstructured*. The unstructured approach is shown schematically in Figure 2(a), where the objective is the identification of the function $\mathbf{f}(\cdot)$ and parameter vector $\mathbf{p}$ that describe how the expression levels of the genes, $\mathbf{x}(t)$, depend on one another and the external inputs, $\mathbf{v}(t)$. Every gene is allowed to interact with every other gene, resulting in at least $N_g \times N_g$ parameters to be estimated from $N_m << N_g$ measurements—a problem that is clearly unsolvable in this ill-posed form. Using various dynamic model types (for example, linear discrete-time (D'Haeseleer et al., 1999; van Someren et al., 2000, Holter et al., 2001), linear continuous-time (Chen et al., 1999), and nonlinear discrete-time (Weaver et al., 1999)), each approach is characterized by the particular device employed to make the problem more tractable. For example, D'Haeseleer et al. (1999) use nonlinear interpolation to increase the number of available data points, while Weaver et al. (1999) reduce the number of model parameters by repeatedly fixing small gene-gene interaction parameters at zero, and then re-estimating the parameters for the new constrained system. Some avoid identifying interactions between thousands of genes by identifying instead interactions between only a handful of gene clusters (Van Someren et al., 2000) or by identifying interactions between composite modes derived from singular value decomposition (SVD) of the data (Holter et al., 2001). Others add sparseness constraints to the network identification problem, selecting models that not only fit the data well, but do so with a minimal number of gene-gene interactions (Van Someren et al., 2001; Yeung et al., 2002). Still others restrict the scope to steady-state system identification through ensembles of small perturbations and the assumption of local linearity, obtaining good results for simulated systems (Kholodenko et al., 2002; de la Fuente et al., 2002; Tegner et al., 2003) and also for a small, well-defined, experimental system (Gardner et al., 2003).

These approaches are novel attempts at solving a very difficult problem, albeit with some questionable underlying assumptions. For example, the biological significance of SVD gene modes regulating other SVD gene modes, or gene clusters regulating other gene clusters, is unclear. While sparseness is an important attribute of any gene regulatory network model, it is not an ideal constraint because there is no guarantee that the sparsest network is the correct one. Furthermore, for systems of realistic scale, there is a combinatorially large number of sparse networks to evaluate. The steady state approaches have the obvious drawback that network dynamics, which may be critical to physiology, are entirely neglected; they also suffer from the possibility that small perturbations from steady state may not be feasible for biological systems, where ultrasensitive all-or-nothing responses are common (Neves and Iyengar, 2002). Finally, these steady state approaches are sensitive to the tradeoff between small perturbations, which are essential for the local linearity assumption to be valid, and measurement noise. Nevertheless, these all represent creative attempts to identify complex systems from a small amount of data and limited prior information.

3.2 A Structured Approach

An intrinsically more tractable approach to gene regulatory network identification is obtained when additional domain knowledge is used to impose mechanistically justifiable structure on the identification problem. For this *structured* approach, there are three levels of structure to

consider: 1) subcellular structure, 2) nuclear connectivity, and 3) dynamical model structure. Structural terms 1) and 2) are unique to the present approach and are discussed in the following paragraphs. Use of dynamical model structure 3) in the present approach is analogous to other studies in which empirical dynamical relationships between system components are postulated and parameterized using experimental data.

With the exception of dynamic regulation of mRNA stability (Wilusz et al., 2001), regulation of gene expression generally occurs through the regulation of transcription initiation (Fickett and Hatzigeorgiou, 1997). Transcription initiation is itself regulated by transcription factors (TFs) that bind to transcriptional regulatory elements (TREs) - short sequences of DNA, 8-24 base pairs in length, in gene promoters where they influence the assembly of the preinitiation complex that initiates transcription. It follows, therefore, that a reasonable first order approximation for the regulation of gene expression is through variation in the complement of active TFs that are present in the nucleus.

Using this domain knowledge, we may first impose *subcellular structure* on $\mathbf{f}(\mathbf{x},\mathbf{v},\mathbf{p})$, shown schematically in Figure 2(b). By subcellular structure, we mean that the overall model from Figure 2(a) has been decomposed into models of the nuclear and cytoplasmic subcellular compartments. The nuclear model, $\mathbf{g}(\mathbf{u}(t), \mathbf{x}(t), \mathbf{p}_g)$, describes how the TFs ($\mathbf{u}$(t)), not the entire complement of genes, regulate gene expression; the dependence on $\mathbf{x}(t)$ is an explicit direct proportionality for first order degradation reactions; and there are roughly $N_g \times (N_t+1)$ parameters, where N_t is the number of TFs in the system ($N_t << N_g$). The cytoplasmic model, $\mathbf{h}(\mathbf{x}(t), \mathbf{v}(t), \mathbf{p}_h)$ (static or dynamic, depending on the available information) describes how the levels of active TFs ($\mathbf{u}$(t)) depend on the expression levels of the genes ($\mathbf{x}(t)$) and the external inputs ($\mathbf{v}(t)$). Assuming that the TFs only need to be expressed to be active (true for *developmental* TFs, Brivanlou and Darnell, 2002), $\mathbf{h}(\mathbf{x}(t), \mathbf{v}(t), \mathbf{p}_h)$ relates levels of the TF mRNA to TF protein levels and is thus very simple, containing $\sim N_t$ parameters. The nuclear model, containing many more parameters than the cytoplasmic model, determines the overall number of parameters in the gene regulatory network model after imposing subcellular structure, of order $\sim N_g \times N_t$. For humans, N_t ~3,000 (Brivanlou and Darnell, 2002), an order of magnitude less than N_g. Thus, by imposing a structure in which only variations in TF expression levels regulate gene expression, not variations in the expression levels of all genes, the number of parameters is reduced by an order of magnitude.

Just as every gene does not regulate every other gene, neither does every TF regulate every gene, and it is thus possible to impose additional structure on the model in terms of *nuclear connectivity*. Nuclear connectivity is defined in the present work as the specification of which genes are regulated by which TFs. Prior knowledge of nuclear connectivity can significantly reduce the number of effective parameters in $\mathbf{g}(\mathbf{u}(t), \mathbf{x}(t), \mathbf{p}_g)$ (and thus the overall model) to roughly $N_g \times N_b$, where N_b is the average number of TFs regulating each gene. N_b is estimated to be of order ~10 for eukaryotes (Arnone and Davidson, 1997). The net effect is an additional reduction of up to two orders of magnitude in the number of parameters to be estimated, thereby rendering the gene regulatory network identification problem even more tractable for modestly sized data sets.

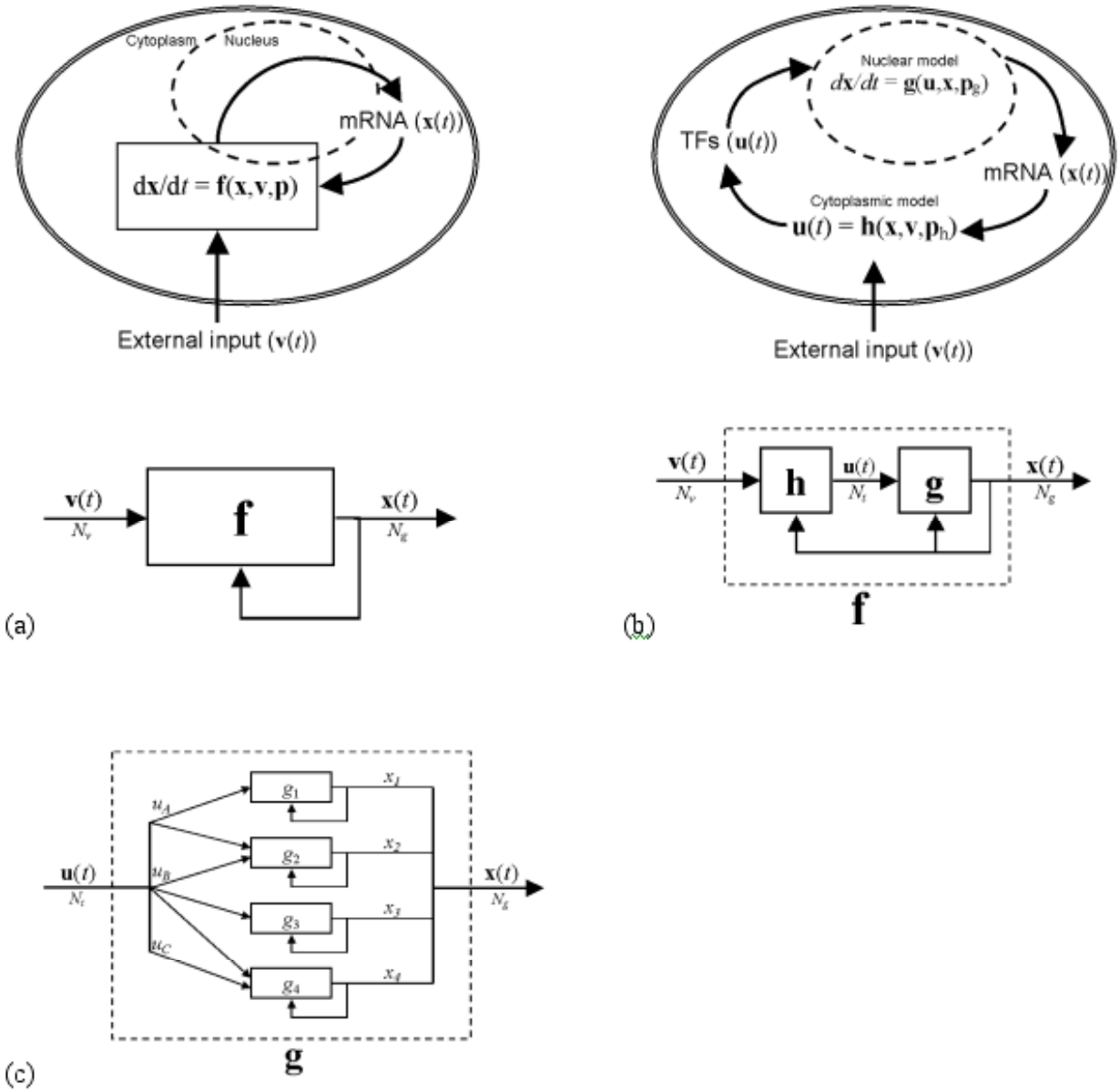

Figure 2: Unstructured and structured approaches to gene regulatory network identification (a) *Unstructured approach*: In this modeling approach, every gene is allowed to regulate every other gene directly or indirectly, with the assumption that analysis will reveal the significant interactions. There are effectively $N_g \times N_g$ parameters or interconnections that must be evaluated, where N_g is the number of genes in the system. (b) *Structured approach, subcellular structure*: In this approach, prior knowledge regarding which genes are transcription factors (TFs) is used to formulate two separate models: (1) A nuclear model that describes how the transcription rate of a specific gene depends on the activity of the TFs, and (2) a cytoplasmic model that describes how the activation of specific TFs depends on the expression levels of the genes and external perturbations. This reduces the number of parameters by an order of magnitude as compared to the unstructured approach. (c) *Structured approach, nuclear connectivity*: Additional structure is imposed on the nuclear model by specifying which genes are regulated by which TFs, reducing the number of model parameters further by up to two orders of magnitude.

While the structured approach can simplify the gene regulatory network identification problem considerably, it also facilitates the inclusion of additional biological complexities when they are known *à priori*. For example, if the TFs are of the *signaling* type, being activated in a complex manner by external inputs ($\mathbf{v}(t)$) and the existing complement of proteins within the cell (Brivanlou and Darnell, 2002), this will be reflected by increasing the complexity in $\mathbf{h}(\mathbf{x}(t), \mathbf{v}(t), \mathbf{p}_h)$ accordingly. Similarly, if the transcription rates of some genes depend in a complex manner on the activities of multiple TFs, $\mathbf{g}(\mathbf{u}(t), \mathbf{x}(t), \mathbf{p}_g)$ will increase commensurately in complexity. The structured approach also makes it possible to identify which particular assumptions have been violated when the models do not adequately describe the data. Note that in the unstructured approach, there is no easy way to include additional complexity or to check for violated assumptions because there are no elements in the framework that reflect the underlying biological mechanisms.

3.3 Data Quantity and Quality

Data quantity and quality are critical issues in model identification in general, but are even more critical for the identification of gene regulatory networks and biological systems overall. At the molecular level, biological systems are more complex and of higher dimensionality (and hence larger in overall problem scale) than chemical process systems, and therefore require more, not less, data for effective modeling and identification. It is difficult, however, to obtain data from biological systems in amounts that are comparable to what is possible with chemical process systems. Gene expression profile time courses, for example, consist of far fewer time points and have significantly more noise corruption (Nadon and Shoemaker, 2002; Sebastiani et al., 2003) than typical dynamic chemical process data. Additionally, perturbations that are routinely implemented on chemical process systems to excite rich dynamic frequencies, such as pseudo-random binary sequences, or the deliberate small amplitude perturbations for exciting only local linear modes, are far more difficult to implement in biological systems, where only dynamically simplistic perturbations (steps and pulses, for example) are generally applied. Nevertheless, the type of perturbation that is used to excite a biological system plays an important role on in how well the underlying gene regulatory network may be identified (Zak et al., 2003a). The structured approach to gene regulatory network modeling, by incorporating biological domain knowledge and permitting the use of a variety of genomic data types, requires the identification of significantly fewer parameters, and is therefore less sensitive to the quality and quantity of gene expression measurements for a given input perturbation. This reduced dependence on measurements of gene expression, however, comes at the cost of dependence on data of different *types*, which introduces into the modeling and identification process another set of challenges and idiosyncrasies unique to each.

Fortunately, gene regulatory network modeling does not occur in a vacuum. Genomic sequences of nearly all model organisms, and many additional organisms, are available in public databases (Baxevanis, 2003). Additionally, functional annotation of the genome sequences is underway (Ashburner et al., 2000; Camon et al., 2003). Functional annotation can specify which genes are TFs, and thus play a role in defining subcellular structure. Finally, the number of tools for predicting promoter regions from genomic sequences (Fickett and Hatzigeorgiou, 1997) and predicting TREs within promoters (Quandt et al., 1995; Liu et al., 2001; Kel et al., 2003) continue to multiply; databases of TREs continue to grow (Matys et al., 2003); and tools that integrate these data in an automated fashion have been developed (Vadigepalli et al., 2003). These tools, which involve predicting TREs in the promoters of specific genes, play a key role in determining nuclear connectivity. In short, for any organism that is likely to be of interest, the means to acquire the suggested prior knowledge already exist, and although it may not be

possible to obtain complete knowledge of nuclear connectivity, what can be obtained will allow significantly more traction on the gene regulatory network identification problem than is possible by using gene expression data alone.

4 A Novel Meta-Clustering Algorithm to Combine the Results from Different Clustering Techniques

Clustering analysis is an important tool to investigate and interpret data. However, there is no perfect clustering approach outperforming its counterparts, and the performance of a clustering method can vary significantly across datasets. Therefore, it is risky to analyze the data based on one particular algorithm. This risk can be alleviated by employing several different clustering approaches, but the challenge then is how to extract a clear picture of the data structure from this clustering ensemble. This paper addresses this challenge by proposing a novel meta-clustering approach to combine different candidate partitionings into one single hierarchical clustering structure. In this framework, different sections of the clustering structures of the candidate partitionings are weighted differently according to how well they reflect the underlying structure of the original data. This is achieved by the calculation, for each candidate partitioning, of a novel distance matrix, defined on the basis of the given clustering structure and the original data distribution. Then, the distance matrices are combined to produce a new clustering structure that provides a better interpretation on the data distribution of interest. Simulations with artificial and real data show that the proposed approach is able to extract the information efficiently and accurately from the input clustering structures.

4.1 Meta-clustering algorithm

As shown in Figure 3, there are three steps in the proposed algorithm. The first step is called alignment. In this step, we transform each candidate partitioning into a format (matrix of cluster-based distances, D_c) that is independent of the given clustering structure. Specifically, entry (i,j) of matrix D_c represents the distance between the vectors providing the probability that the data points i and j belong to every defined cluster in the given partitioning (i.e., the distance between the vectors of soft membership for data points i and j).

The consistent format of the D_c matrices facilitates the integration process in the combination step, which consists on the average of the D_c matrices of all the candidate partionings. The underlying assumption of this scheme is that data points belonging (nonbelonging) to a natural existing cluster have a low (high) cluster-based distance in all the candidate D_c matrices, while only dubious cases will get intermediate values of the cluster-based distance in some D_c matrices. This attractive property of the cluster-based distance is a consequence of the use of the original data attributes in its calculation.

Finally, in the reclustering step, the proposed algorithm extracts a hierarchical clustering structure from the average D_c matrix obtained in the combination step. After the reclustering step, we obtain a hierarchy of clustering structures. If the desired number of clusters is known, we can use this knowledge to pick up the corresponding partition from the resulting tree. On the other hand, the merging cost at each step provides hints for selecting this number.

4.2 Experimental results

To simplify the discussion, we first assume that the desired number of clusters is known. All the clustering approaches have been applied on six datasets, which include three artificial datasets, two machine learning benchmark datasets (iris and wine datasets) and one dataset collected from a biological application (the expression level of 800 cell cycle related genes in budding yeast Saccharomyces cerevisiae measured during different cell-cycle stages and categorized into four different functional groups: chromatin structure, glycolisis, protein degradation, and spindle pole). The datasets are selected so that the desired partitioning for each dataset is known. This external information provides a “golden” rule for the evaluation of the clustering results.

Specifically, we use the Rand index to measure the difference between an obtained clustering structure and the desired partitioning. The Rand index is computed by examining all pairs of data elements in the dataset after clustering. If two data elements belong to the same cluster in both the desired partitioning and the obtained clustering structure, this counts as an agreement. If two data elements are in different clusters in both the desired partitioning and the obtained clustering structure, this is also an agreement. Otherwise, there is a disagreement. The Rand index is computed by dividing the number of agreements by the sum of agreements and disagreements. Thus, it is a measure of how closely a clustering result matches the desired partitioning (that is, it is a measure of clustering accuracy). The value of the Rand index falls in the interval [0, 1], with 1 representing a clustering result that perfectly matches the desired partitioning.

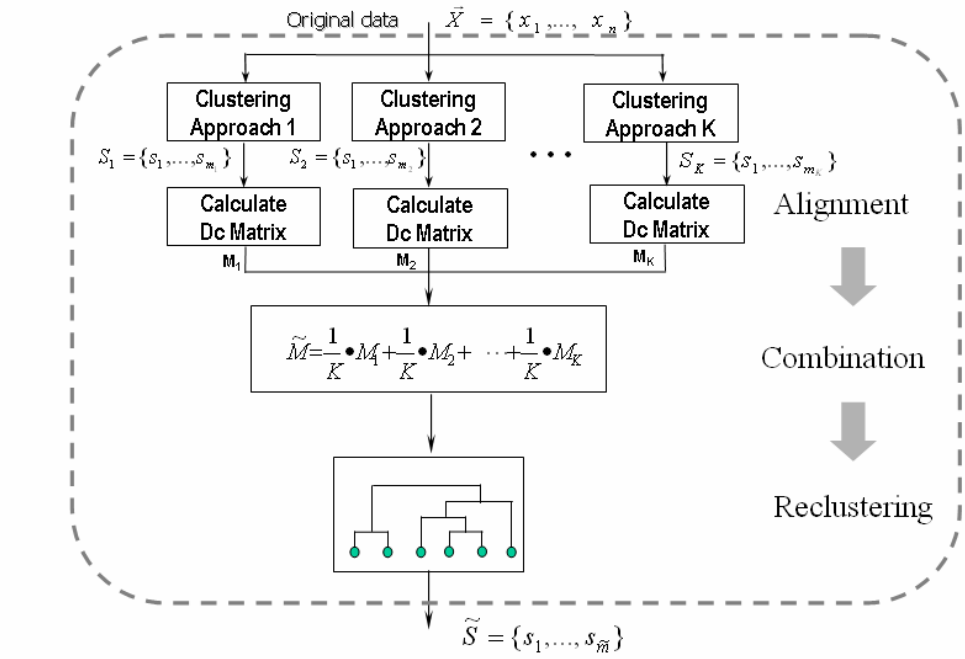

Figure 3: Meta-clustering approach to combine different clustering results.

Table 1 shows the values of the Rand index for different clustering results on the six datasets. For each dataset, the clustering results are divided into four groups: the group of candidate clustering structures, the clustering selector group (where each selector chooses a candidate clustering according to a cluster validation index), the group of previously existing clustering ensemble techniques (HGPA, CSPA, and MCLA), and finally the proposed meta-clustering algorithm. As shown in the table, for the first two datasets the desired clustering results are obtained by one of the candidate approaches (SOM for the first dataset and linkage for the second). In these two cases, the CSPA and the proposed meta-clustering approaches are both able to extract the correct partitioning among all the candidates. For the other four datasets, there is no candidate partitioning reflecting the desired grouping perfectly. As shown, the results provided by the meta-clustering method on all these datasets are better than any of the candidates. Moreover, the proposed approach outperforms all the other methods, except for the MCLA approach in the last dataset, which produces exactly the same result.

Table 1. Performance of all the methods considered here when applied on the six datasets

	Test1	Test2	Test3	Iris	Wine	Yeast
K-means	0.5	0.5501	0.8432	0.8805	0.9125	0.9063
SOM	1	0.5347	0.8564	0.7805	0.9467	0.9479
Linkage	0.5	1	0.3500	0.8930	0.7202	0.8414
Selector (DB-index)	0.5	0.5501	0.8564	0.8805	0.9467	0.8414
Selector (Silhouette index)	0.5	0.5501	0.8432	0.8930	0.9125	0.9479
HGPA	0.5	0.5	0.5527	0.7857	0.5898	0.8064
CSPA	1	1	0.8470	0.8632	0.8528	0.8457
MCLA	0.6043	0.6860	0.9173	0.8805	0.9125	0.9815
Meta-clustering	1	1	0.9301	0.8995	0.9623	0.9815

We now discuss how to decide the number of clusters in the proposed meta-clustering approach. As described before, the last step of the proposed algorithm is an agglomerative hierarchical clustering. Therefore, independently of the clustering techniques used as candidate inputs, the meta-clustering approach always provides a dendrogram that groups the data elements into different number of clusters in a hierarchical way. The merging cost values reflect the structural change in the clusters obtained in each step, and they can be used to estimate the number of clusters: Merging in the single linkage algorithm can be stopped when the merging cost presents a sudden large increase with respect to previous merging costs. Provided that the inherent classes are well separated, this method can detect the optimal number of clusters accurately.

Figure 4 shows the dendrograms produced by the proposed approach when applied on the six datasets introduced before. Notice that the merging cost in the first dataset stays low until the number of the clusters reaches two. Then, the last merging has a cost much larger than before, which indicates that the desired number of clusters is two (which agrees with the desired result). As shown in Figure 4(b-e), similar results can be found for the other two artificial datasets and for the two machine-learning benchmark datasets, where the desired numbers of clusters are correctly suggested in the proposed meta-clustering approach by considering the jumps in the merging cost.

One exception in the relationship between merging costs and number of clusters can be found in the result from the yeast data, which suggests eight clusters instead of four clusters, as shown in Figure 4(f). This occurs because there are two classes that are separated (each) into more than one cluster. For instance, the class “Glycolysis” is divided into four clusters. The reason is that all the candidates failed to group the profiles in this function class together in one single cluster. Therefore, although in the next three steps of the meta-clustering these four clusters are correctly merged together (results not shown), the merging costs are pretty high, which means that there is little support for these mergings from the candidate clustering results. This example shows that the number of clusters estimated by the proposed meta-clustering method is heavily dependent on the quality of the input.

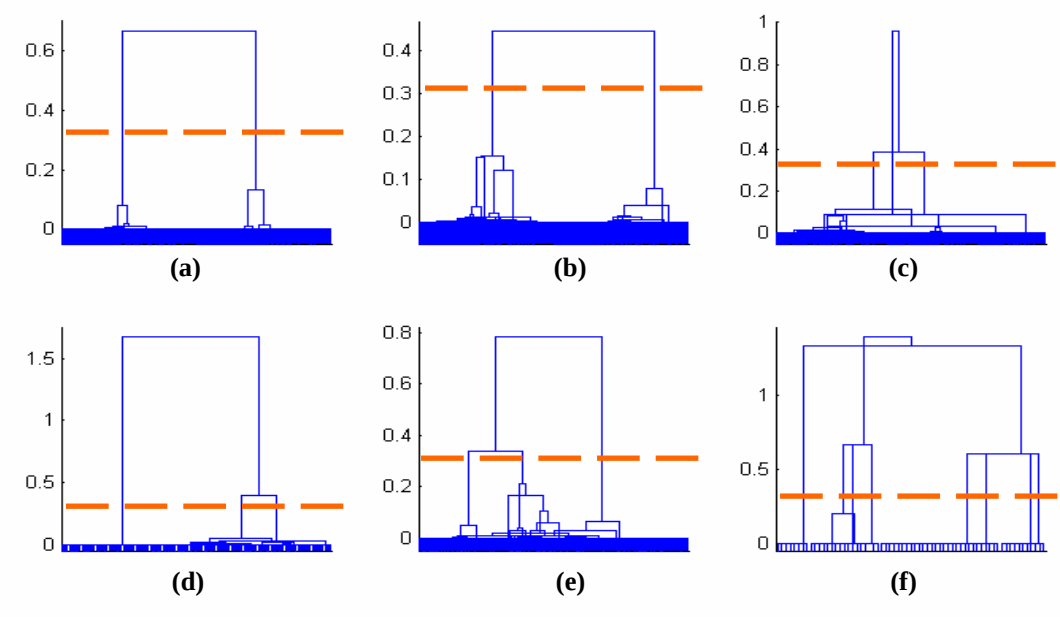

Figure 4: Dendrograms produced by the meta-clustering approach on the six datasets, where the suggested partitioning in each dendrogram is marked by a dashed line. (a) From artificial dataset test1; (b) from artificial dataset test2; (c) from artificial dataset test3; (d) from the iris dataset; (e) from the wine recognition dataset; (f) from the yeast dataset.

5 PAINT: Promoter Analysis and Interaction Network Toolset for gene regulatory network structure prediction

5.1 PAINT Architecture

The modular architecture of PAINT is not organism-specific. The key requirements are the availability of annotated genome sequence and information on transcription factor binding site motifs. PAINT 3.3 can conduct analysis specific to the mouse, human and rat. The toolset is constituted of the following four components:

1. **Preprocessor**: A Perl module that builds the PAINT promoter database based on Ensembl and Unigene annotation that can be queried using the Ensembl GeneID, Locus Link, Unigene ClusterID or Clone ID (GenBank accession number). This allows for faster processing at runtime in fetching the promoter sequences.
2. **Upstreamer**: A Perl module that provides the functionality of sequence retrieval from the UpstreamDB database given a list of unique identifiers for the genes of interest.
3. **TFRetriever**: A Perl module that processes the retrieved sequences through the transcription factor inspection/discovery programs. The dynamic nature of the databases containing transcription factor information and user-specified parameter options require online retrieval rather than an offline processing for all the promoters in UpstreamDB.
4. **Analysis and Visualization**: A Perl and R module that contains functions for analysis and visualization of CIM. A matrix image with optional clustering of data and a network layout diagram are available. Also, produced are various file formats: SBML and GraphML for use in JDesigner (Hucka et al., 2001), Cluster/TreeView (Eisen et al., 1998), Pajek (Batagelj & Mrvar, 1998) and Cytoscape (Ideker et al., 2002). These can be used in the subsequent network analysis and visualization.

The modular architecture of PAINT is depicted in Figure 5. A detailed description of each of the modules and the input-output relationships is presented next. A discussion of the issues involved and specific choices made in the tool development is presented in the Discussion section.

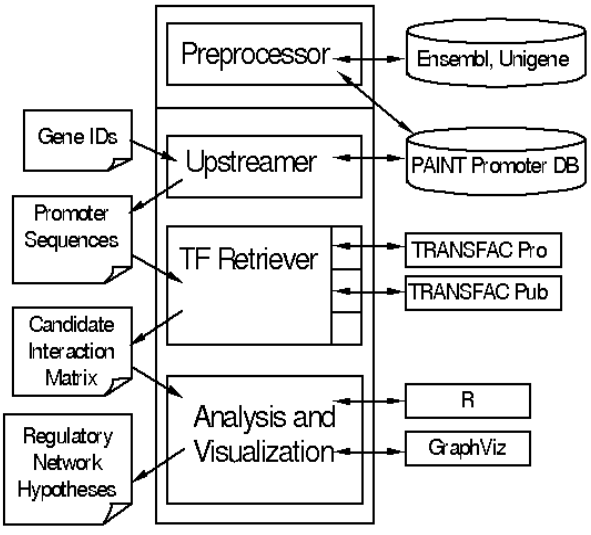

Figure 5: A schematic of the PAINT workflow indicating the data types on one side and the resources used on the other. The resources can either be local or accessed on the network. The input and output data formats are usually text-based so as to allow processing by other tools and spreadsheet-like software if necessary. This modular architecture makes it easy to plug in updated/new modules as well as interact with external utilities.

5.2 PAINT Modules

The UpstreamDB module. For an organism of interest, the principal requirement for constructing the promoter database is annotated genome sequence assembly. Several genome assemblies are available for mammalian systems, for example, Ensembl (Hubbard et al., 2002), Santa Cruz (http://genome.ucsc.edu), Celera (http://www.celera.com). The UpstreamDB database was constructed for all the annotated genes (known and putative) in the Ensembl genome database for Mus Musculus. For each gene, 5,000 base pairs (bp) upstream (5' to the gene), the first exon of the open reading frame (ORF), and 100 base pairs downstream (3' to the end of first exon) were retrieved from the genome and placed in a temporary database (TempDB) prior to the identification of the Transcription Start Sites (TSS). In the case of genes for which upstream sequence of length 5,000 bp is not available in the genome database (due to assembly being incomplete), the maximum available sequence was retrieved. The retrieved sequence was placed in the database only if at least 300 bp sequences immediately 5' to the gene were available. The genome database contains sequences in 5' to 3' orientation on a single strand (conventionally denoted as +1) of DNA. For the genes that are located on the strand –1, the sequence from the genome database was reversed and complementary base pairs were computed to produce the upstream sequences.

The key aspect of the analysis is using the correct sequence to represent the cis-regulatory control regions. Note that this requires information about the 5' untranslated region (UTR) of each gene in order to correctly identify the TSS and hence the corresponding cis-regulatory control region for each gene. The first exon of a gene as annotated in the Ensembl database does not necessarily correspond to the TSS (Davuluri et al., 2001). This creates difficulty in identifying and retrieving the appropriate sequence data corresponding to the cis-regulatory region for the genes of interest. A sequence-driven approach can be employed for computing the TSS of a given gene by alignment with corresponding expressed sequence data, for example, EST sequence (dbEST) or cDNA sequence (from GenBank). Of particular interest to the mouse model system is the effort designed to provide 5' end data for mRNAs (RIKEN clone sequences: Kawai et al., 2001). These clone sequences were aligned to the gene upstream sequence in the TempDB to estimate the TSS for each gene. The sequence alignment program Megablast was used with option '-W48' to use a word size of 52 in alignment. For each gene, the corresponding alignments were filtered by the following criteria:

1. The alignment should have less than 3 mismatches.
2. The alignment should be closest to the 5' end of the aligned clone sequences.
3. If 5' ends of multiple clones align well, then the alignment that is 5' most on the gene upstream sequence is selected.

The position on the gene upstream sequence that is marked by the above alignment and filtering was considered to be the estimated TSS. Using this procedure, an estimate of the TSS's for 5040 genes was obtained. For the remaining genes, a TSS prediction tool Eponine (Reference servlet.sanger.ac.uk:8080/eponine) was employed to come up with an estimate of TSS within the 5000 bp 5' from the start of the ORF. The TSS's for 2278 genes were identified in this manner. For the remaining 14,045 genes, the start of the ORF was considered as the TSS. After the TSS's for genes are estimated for each gene, an updated upstream sequence of 2,000 bp 5' to the estimated TSS was retrieved and stored in the UpstreamDB database.

In addition to the promoter sequence for each gene, UpstreamDB also contains the cross reference tables that enable retrieval of promoters using Unigene ClusterID, LocusLink, and the

cDNA clone Accession number. This cross reference was constructed using information from the Unigene database. This allows for convenient retrieval of the promoter sequences directly from a list of genes marked as significantly varying in expression by the microarray analysis software or other gene expression analysis methods.

The Upstreamer module contains Perl functions that can be wrapped for inclusion in UNIX shell scripts, Perl scripts, web-based scripts such as PHP, and Open Agent Architecture (for use as a BioSPICE module). The input from the user is a list of identifiers for the genes of interest and the number of base pairs of the upstream sequence needed for analysis. The length count is from the start of the gene toward the upstream (5') end. However, the retrieved sequence is written from 5' to 3' direction as per convention. The output of the module is the upstream sequences of specified length for the genes that are referenced in the UpstreamDB database. The output is in FASTA format for further processing by transcription binding motif inspection/discovery software.

The TFRetriever module is envisaged to contain several sub-modules that can communicate with various local and web-based motif inspection and discovery software such as MatInspector (Quandt et al., 1995) and MEME (Bailey et al., 1994). A motif is a characteristic sequence of a binding site and functionally similar motifs are grouped together into families. The set of vertebrate transcription factor families is utilized for promoter inspection. The output of the TFRetriever module is the output from the motif discovery program for each input sequence list. PAINT until v2.3 (as used in the case studies 1 and 2 below) contained the sub-module for interacting with MatInspector software. The current version PAINT 3.3 (as of January 2006) contains only the sub-module for interacting with MATCH software (Kel et al., 2003), in conjunction with either TRANSFAC Professional or TRANSFAC Public versions (Matys el a., 2003). This is the version currently supported on the Dashboard 6.0 used in case studies 3 and 4.

At present, FeasNetBuilder, a submodule of the TFRetriever, can process the output from MATCH to construct an interaction matrix representing a candidate set of connections in the regulatory network based on the promoter sequence and TF/TRE information. In the set of promoter sequences processed, the complete list of TREs was generated. The columns of the interaction matrix correspond to the TREs and each row corresponds to a gene from the input list. If the parameter for binary counting is set in PAINT, the regulation of a gene is represented by a 1 if the corresponding TRE is present on the promoter for that gene, and by a 0 otherwise. This matrix represents the constraints to a network identification scheme. The interaction parameters corresponding to zeros in the candidate matrix need not be computed, substantially reducing the dimensionality of the identification problem. If the parameter for binary counting is not set, each element of the CIM will be equal to the number of corresponding TREs found on the respective promoter.

The FeasnetBuilder module contains a submodule named StatFilter that computes p-values for the over-representation of the TREs in the set of promoters considered with respect to a background set of promoters. Specifically, the p-values give the probability that the observed counts for the TREs in the set of promoters could be explained by random occurrence in the background set of promoters. The p-values are calculated using the hypergeometric distribution (Bury, 1999; Jakt et al., 2001; Elkon et al., 2003). Typically, the reference set is that of the genes on the microarray utilized in the experiments. For each TRE V$X, given (1) a reference CIM of n promoters of which l promoters contain V$X and (2) a CIM of interest with m promoters of which h contain V$X, the associated p-value for over-representation is given as equation 2:

$$p = \frac{\sum_{i=h}^{m} \binom{l}{i} \binom{n-l}{m-i}}{\binom{n}{m}} \quad (2)$$

The p-value for under representation of a TRE in the observed CIM is calculated similarly with the summation in the above equation going from 1 to m. These estimates of significance can be utilized in filtering for those TREs that meet a threshold (say, $p < 0.1$) to identify most likely regulators of the genes considered in the experimental context of interest. For the case studies presented here, the CIM corresponding to the ~3200 annotated cDNA clones on the microarray utilized for experiments was considered as the reference CIM. Given no information about the source of the genes from which the input list to PAINT is generated, PAINT can optionally utilize the CIM corresponding to all the genes in the UpstreamDB database as a reference CIM.

The Analysis and Visualization module contains various functions for the visualization and analysis of the CIM. An image of the interaction matrix is produced in which the individual elements of the matrix are represented by a color based on the significance values for that particular TRE (p-values for over-representation in the observed CIM). This module also contains functionality for hierarchical clustering using 'R' software for statistical analysis (http://www.r-project.org). For clustering, the pair-wise distance that is most appropriate for the CIM data is the binary distance. The binary distance between two genes (or TFs) can be computed as the ratio of number of elements for which the two rows (or columns) are dissimilar to the total number of elements for which either of the rows contains a 1. For the genes, binary distance is the dissimilarity between the regulatory patterns of two genes as related to the total number of distinct binding sites present on either of them. For the TFs, binary distance is the dissimilarity between the regulatory patterns of two TFs as related to the total number of genes either of the TFs can regulate. In PAINT, the clustered data can be visualized as a matrix layout with the hierarchical tree structure aligned to the rows and the columns of the CIM. The zeros in the matrix are shown in black and the non-zero entries in the CIM are colored based on the p-value of the corresponding TRE. The brightest shade of red represents low p-value (most significantly over represented in the CIM). Conversely, the brightest shades of cyan represent smaller p-values for under representation in the observed CIM indicating more significantly under represented TREs. This image can optionally represent the cluster index of each gene, where such cluster indices are generated from other sources such as expression or annotation-based clustering. With such visualization, it is straightforward to explore the relationship between expression/annotation-based clusters and those based on cis-regulatory pattern (i.e., CIM). The Analysis and Visualization module can also generate histograms of the network connectivity from the CIM to provide additional insights into the regulatory network of interest. A histogram of the sum of all columns in the CIM provides the distribution of the number (or fraction) of TREs that can regulate a given gene. This distribution is typically uni-modal with long tails indicating that very few genes are regulated by very few or very many TREs.

Similarly, a histogram of sum of all the rows in CIM provides the distribution of number (or fraction) of genes that are regulated by a TRE. Typically, this distribution is monotonically decreasing indicating that most of the TREs are present on few genes each (fine-tuned regulation) and relatively few TREs are present on large number of genes (system-wide effects). The Analysis and Visualization module can also generate a network layout diagram using the GraphViz libraries (available at http://www.research.att.com/sw/tools/graphviz/).

6 BioSPICE Software: TJU Contributed Modules

In this section we present a detailed description of the BioSPICE modules developed for gene regulatory network analysis. Some of the modules presented here were supported in Dashboard version earlier than 6.0, and were later subsumed by a completely reworked PAINT module. However, descriptions of the currently unsupported modules are also included here to present the major software development efforts for the entire duration of the project. The details of the current PAINT module are presented after the description of previous versions of the tools.

BIOSPICE Software module contribution (February 2004)

6.1 MetaCluster Toolbox

Input: Microarray expression data (Timeseries format).
Output: Cluster membership for each identifier (SBML2).

A diverse supply of clustering algorithms are available for data analysis, all of which generate results which are specific to the algorithm used or even to the individual iteration. MetaCluster was developed to help biologists mine several different clustering results for those data relationship insensitive to the clustering methods used. MetaCluster provides a computational tool which co-analyses diverse clustering results to highlight the relationships that are stable across algorithms. These method-independent co-clustering results provide the strongest evidence for biological significance. The MetaCluster Toolbox included in this Dashboard contribution features a GUI that takes Timeseries as input (such as microarray expression data), allows the user to cluster the data interactively, and produces a SBML2 output of the clustering results.

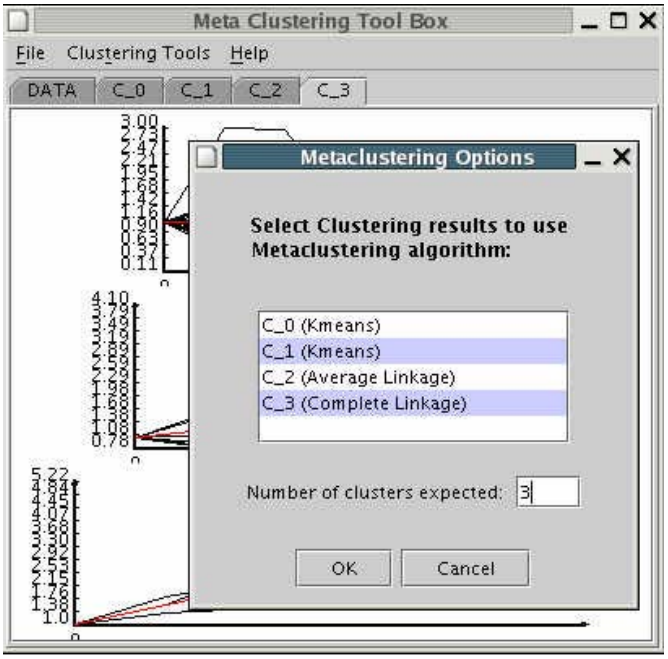

Figure 6: Merging of multiple clustering results (for robust clustering of gene expression data) provided by MetaCluster Toolbox.

On the Dashboard, MetaCluster Toolbox starts up the GUI as shown in Fig 6, with the data obtained from the workflow in the Timeseries format. Currently, the MetaCluster Toolbox implements the clustering algorithms as shown in the Figure 6. The user can choose any of the clustering algorithms, and specify the number of clusters expected. Each choice of clustering opens a new tabbed pane of clustering results as shown in the Figure 6. Using MetaCluster algorithm requires at least two clustering results. If the user selects Meta-clustering from the Clustering Tools menu, the MetaCluster Options dialog box is presented as shown in the Figure 6. This way, the user can continue with clustering the data with different algorithms/options until a satisfactory result is obtained. After a reasonable clustering result is obtained, the user may choose to quit the MetaCluster Toolbox and pass on the clustering results to the next component in the workflow. To do this, the specific tabbed pane in which desired results are present has to be selected (for example, C_4 in the Figure 6). At this point, choosing Output to SBML2 and

Quit option from the File menu quits the MetaCluster Toolbox, and sends the clustering results out. Please note that closing the MetaCluster Toolbox window forcibly does not send out the clustering results, and may result in abnormal termination of the workflow. The cluster information is stored in the <annotation> node of the <species>, where species is the gene identifier from the Timeseries data. Within the annotation, the value attribute in <dbi:user-def name="metacluster"> indicates the cluster membership. Please refer to the sample output file below for specific details.

6.2 CloneUpdater
Input: Gene list, with optional annotation (SBML2).
Output: Gene list, with new/updated annotation (SBML2).

Gene annotation is one of the most important data classes used in the post-genomic era. CloneUpdater provides the biologist an easy to use path to the most "up to date" annotation of genes and associated reagents such as EST clones. It is compatible with many different types of identifiers and provides a large number of options for updating preexisting annotations or adding new annotations to user-provided identifiers from many different databases including UniGene, LocusLink, RefSeq, and more. Processing more than 50 identifiers per second, CloneUpdater is fast enough to be used for one gene identifier or tens of thousands. In addition, CloneUpdater has the important ability to find identifiers that represent the same gene and thereby highlight redundancies in large reagent collections. In the context of analyzing gene regulatory networks, CloneUpdater can be used as a preprocessor for PAINT: it can eliminate redundant clones, and update annotations that PAINT can process. This Dashboard release of CloneUpdater features a GUI that offers the same features as the online version located at: http://www.dbi.tju.edu/cloneupdater.

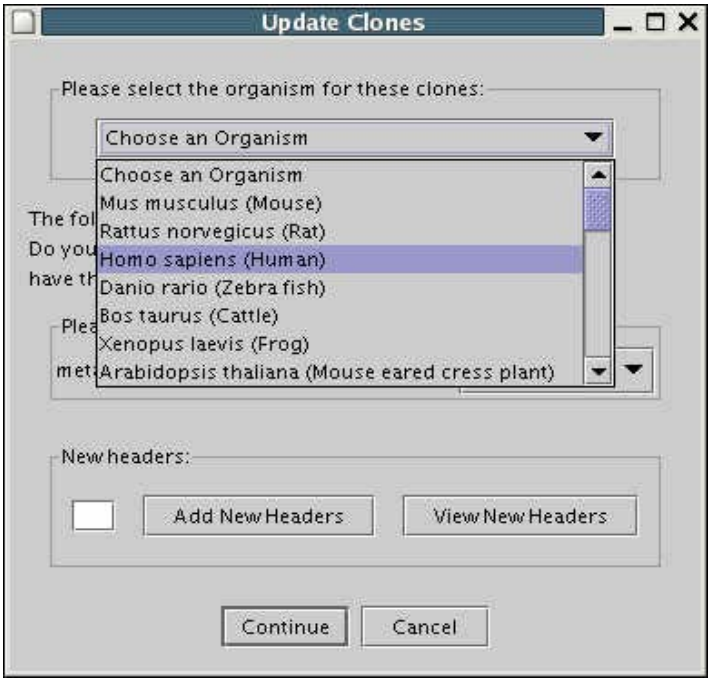

Figure 7: Updated clone annotation data (based on latest UniGene release) provided by CloneUpdater.

CloneUpdater starts up with the GUI as shown in Figure 7. By this point, the SBML2 input is parsed for existing annotation, if any. The next step is to choose the organism (to which the clones belong to) using the Choose an Organism drop down list in the main window (Figure 7(a)). At this point, the user is presented with the option of fetching more annotation information (from UniGene database) for the selected clones, using the Add New Headers button. The user is expected to enter the number of desired new annotation elements in the box provided, and then click the Add New Headers. The drop down list in the Define New Headers dialog box lets the user choose desired annotation information for each of the clones in the list. After the desired annotation is chosen, clicking on the Continue button submits the request for processing. It takes about 3 seconds to process a list of 30 clones. A drop down list lets the user to select All Clones,

Distinct Clones (default), Redundant Clones or Clones not in UniGene to be sent to the next component in the Dashboard workflow. Clicking the Output to SBML2 and Quit button quits CloneUpdater and transfers the control to the Dashboard. In the output file, for each of the <species>, <annotation> tags are added or updated as specified.

6.3 Promoter Analysis and Interactive Network Toolset (PAINT)

PAINT was developed to provide the biologist a computational tool to integrate functional genomics data, for example from microarray-based gene expression analysis, with genomic sequence data to carry out transcriptional regulatory network analysis (TRNA). TRNA combines bioinformatics, used to identify and analyze gene regulatory regions, and statistical significance testing, used to rank the likelihood of the involvement of individual transcription factors, with visualization tools to identify transcription factors likely to play a role in the biology under study. In addition this tool can output results in several different formats for use with modeling and simulation tools. The project is conceived and implemented as an automated modular scalable, extensible, integrative framework of software tools. PAINT's modular architecture, in combination with BioSPICE's Dashboard and OAA framework, allows a biologist to take advantage of the existing, as well as future BioSPICE Agents. PAINT is also available online at: http://www.dbi.tju.edu/dbi/tools/paint

The Dashboard release of PAINT (as of February 2004) consists of the following modules:

1) PAINT Feasnetbuilder: Takes an annotated gene list in SBML2 format and produces an output of genes and Transcriptional Regulatory Elements interaction.

2) Feasnet Adapter: Converts genes and TREs interaction in SBML2 format to Feasnet Object that can be used by PAINT FeasnetViewer.

3) PAINT FeasnetViewer: Takes a Feasnet Object of interest, and optionally a reference Feasnet to analyze the relative significance of TRE occurrence in the gene list of interest against the given reference. The module has a visualization component, and an output to the PtPlot module.

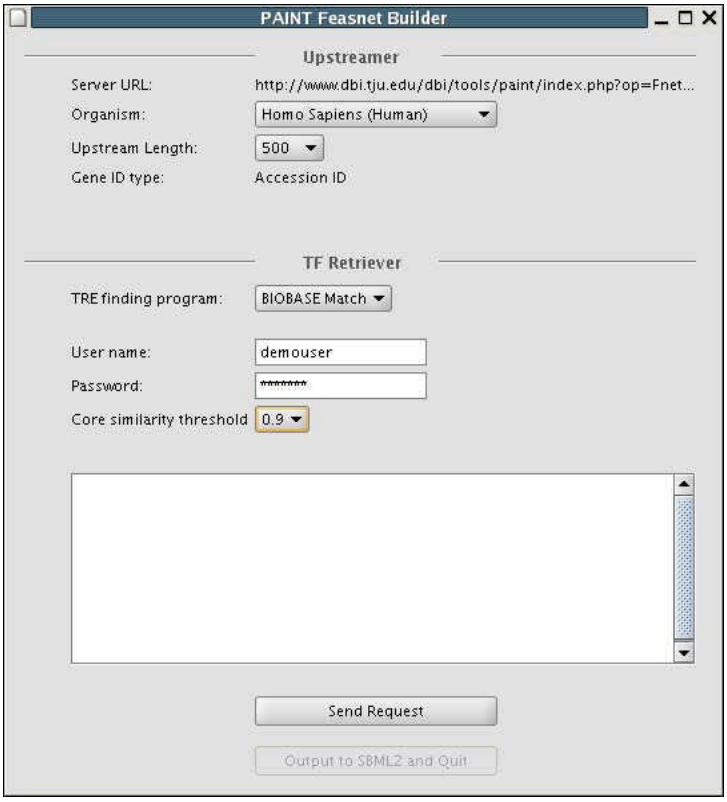

Figure 8: Promoter sequences gathered and construction of a Candidate Interaction Matrix (CIM) based on the TREs present on the sequences are provided by the PAINT Feasnet Builder.

<u>PAINT FeasnetBuilder:</u>

Input: Annotated gene list (SBML2).
Output: Gene-TF interaction data (SBML2).

PAINT Feasnet Builder starts up with the GUI as shown in the Figure 8, with the clone list received from the workflow. PAINT currently maintains a promoter database for Mouse, Human and Rat. In the Organism select box, the user is expected to choose the organism to which the input clones belong to. This module uses TRANSFAC Professional by Cognia Corporation to find known

TREs. The user is expected to be registered with Cognia.com to be able to use the BIOBASE Match. For our example, we chose an Upstream Length of 500 basepairs, and Core similarity threshold of 0.9. Clicking on the Send Request button posts the request to the web-based PAINT. This process takes about 30-40 seconds to complete. The Output to SBML2 and Quit button is activated as soon as the processing is complete. Clicking this button closes PAINT Feasnet Builder and sends the gene-Transcription Factor data to the downstream module on the Dashboard. In the output, each TRE on a gene is represented as a <reaction>, with Transcription Factor in the <listOfReactants>, and gene identifier in the <listOfProducts>.

PAINT Feasnet Adapter:
Input: Gene-TF interaction data (SBML2).
Output: Gene-TF interaction data (Feasnet Object).
Feasnet Adapter module converts gene-TF interaction in SBML2 format to Feasnet format that the Feasnet Viewer module can use. This operation doesn't require any user interaction.

PAINT Feasnet Viewer:
Input: Gene-TF interaction data of gene list (Feasnet), Gene-TF interaction data of reference (Feasnet) (optional), and clustering information of gene list (SBML2) (optional).
Output: TRE over-representation/under-representation, relative to the specified reference (PlotML).
Feasnet Viewer module is the analysis and visualization component of the PAINT. Currently, the module can:
1) Analyze the significance of the TREs present-TRE over/under representation-for each cluster in the gene list, relative to a reference (such as all the genes in a microarray experiment, or entire genome).
2) Display the Gene-TF interaction as a "matrix".
3) Export the five most significantly over-represented or most significantly under-represented TREs across all the clusters.

Feasnet Viewer starts up with the GUI as shown in the Figure 9. Clicking the View Feasnet image button computes analytical p-values for each TRE occurrence on the given genes relative to the specified reference, and displays the image in a tabbed pane as shown in the Figure 9. For these significance values to be meaningful, note that the intended reference gene list should also use the same parameters for finding TRE occurrence as the desired gene list.

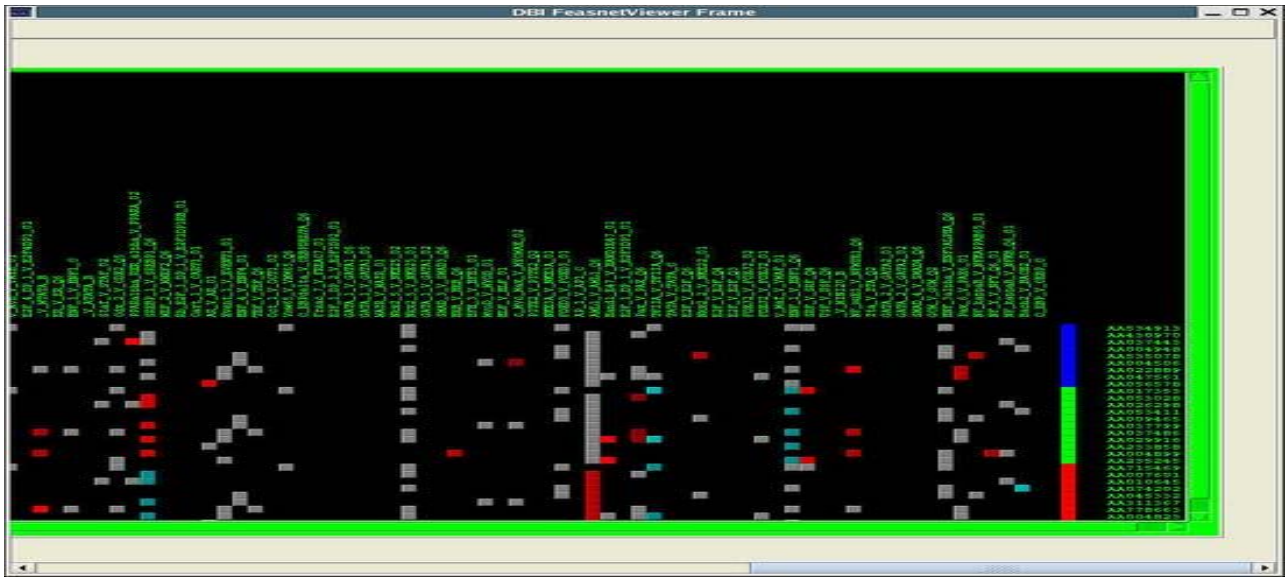

Figure 9: PAINT FeasnetViewer: visualization of the TRE occurrence on the promoter sequences of genes are color coded based on statistical significance. Gene Identifiers are indicated by row labels on the right hand side and the TRE Identifiers are indicated by column labels on the top. Clusters containing each gene are also shown (left of the row labels).

In the Feasnet image, each "dot" indicates the presence of the specific TRE (along columns) on the promoter of the given gene (along rows). Over representation is indicated by different shades of red (the brighter the red color, the greater the TRE is over represented), and under representation by cyan. The gray shades mean that the TRE occurrence in this gene cluster is not significantly different from a randomly picked gene cluster. The colored bar to the left of the gene identifiers indicates the cluster membership (generated from MetaCluster Toolbox). After the image is generated, clicking on Export to PtPlot button prepares the statistics of five most over/under represented TREs across all the gene clusters. Closing the window sends the PlotML data to the next component (typically, PtPlot module) on the Dashboard. Typically, the significance of a TRE varies across the gene clusters. Figure 10 shows the significance score (log scale of probabilities) of over/under-represented TREs across all the gene clusters. A positive value in the plot indicates over-representation, and negative value indicates under-representation. For example, the TRE COREBINDINGFACTOR_Q6 (blue color bars) is significantly over-represented in cluster 3 (significance score ~1.5), compared to the other two clusters (significance scores < 0). An example TRNA workflow is shown in Figure 11.

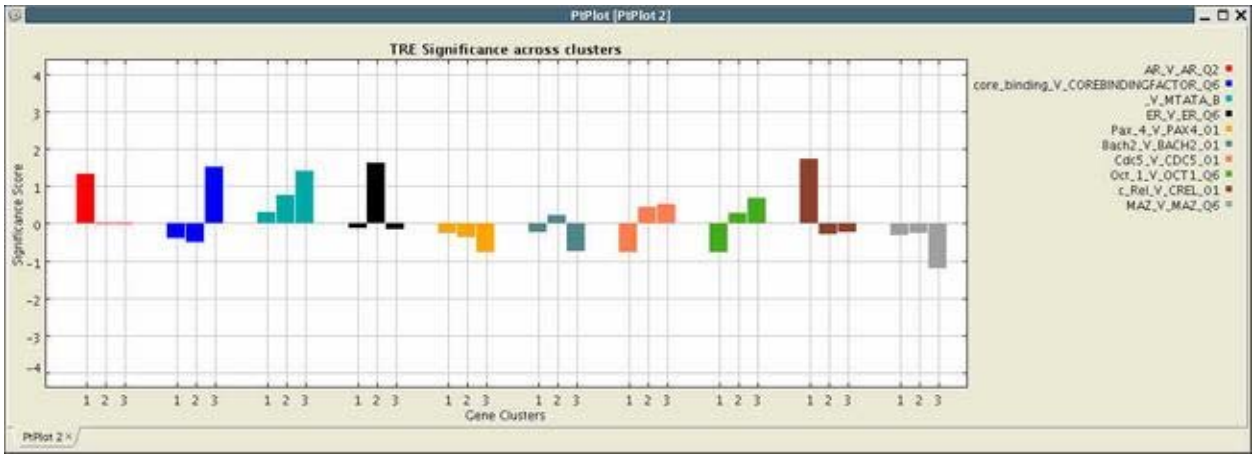

Figure 10: PtPlot module indicating TRE significance scores, filtered by the p-Over < 0.10

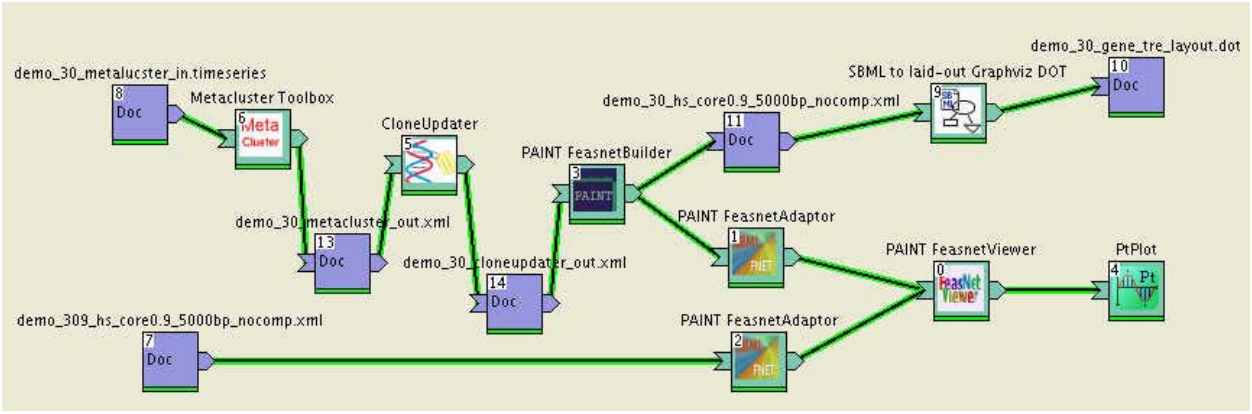

Figure 11: An example Dashboard workflow using TJU modules for gene regulatory network analysis (February 2004). Note that on current Dashboard, only PAINT module is supported. The PAINT module on Dashboard 6.0 has been reworked with a new GUI and Input/Output formats to interoperate with NCA and KAGAN.

6.4 Updated PAINT Module (since June 2005)

This PAINT module release features a complete reworking of PAINT with improved UI, and the underlying framework (Figure 12). Currently, PAINT runs on Linux, Windows and Mac OSX, while the Mac OSX version is not supported under BioSPICE because of Java incompatibility issues on MacOSX. The primary design objectives of this architecture are: Extensibility, Scriptability, and Simplicity. Extensibility refers to the aspect where components can be added to the core with relative ease. Though the typical usage is through the GUI, it is possible to construct PAINT Analyses entirely through programming. One of the future enhancements is to be able to load and save Analyses. The proposed analysis file format allows a user to construct PAINT Analyses outside the GUI. Some of the key concepts in the new interface are described below.

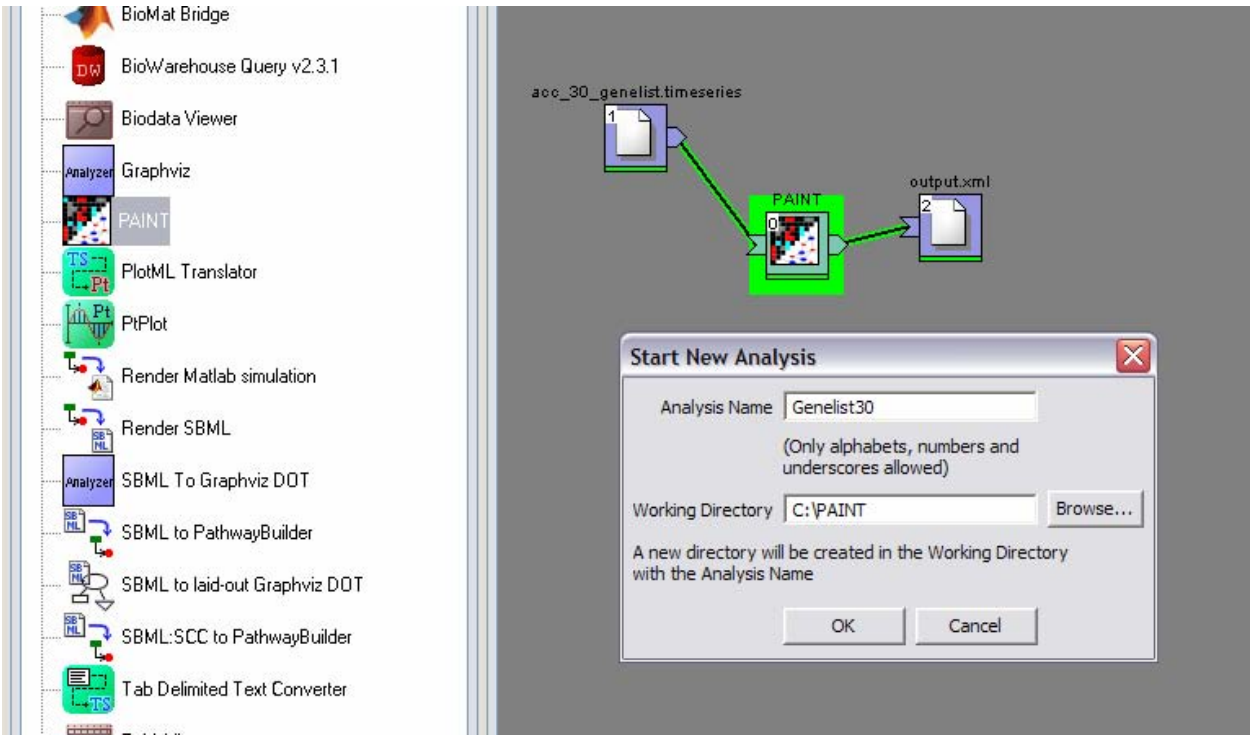

Figure 12: PAINT Workflow, and the New Analysis dialog on the current Dashboard

Analysis Tree: An Analysis is like an interactive workflow, where the user can add Components to the workflow, with the ability to review the results of each Task. An Analysis consists of predefined categories of Components, and the history of the Components - what parameters were used to generate the components. In other words, an analysis includes all the necessary information to reproduce all the components in the analysis. PAINT Analysis is organized as a tree like structure, referred to as the Analysis Tree. At the root level lie the Analysis parameters, followed by the Component categories as the children. Further, each component category can hold multiple instances of the component, each generated with a (possibly) different set of parameters. For example, the Component category Promoter Sequences may contain promoter sequences of length 2000 base pairs and 5000 base pairs. This organization enables the user to work with different parameter combinations in the same Analysis.

Task: A Task is the process that generates an Analysis Component. Each task has an associated GUI component that is designed to receive user input for that Task. Executing a Task is the only way to add a Component to the Analysis. Each Analysis has an associated list of Tasks, which determines what components that analysis can have. Depending on the state of the Analysis, only some of the Tasks may be available. In other words, only those Tasks are enabled, whose dependent Components exist. For example, Finding TREs on promoter Sequences is disabled until the Analysis has a Promoter Sequences component. There may be other limitations too, such as, Import Gene List Task being disabled, if there is already an existing Gene List component.

Analysis Component: An Analysis Component is a part of an Analysis, and consists of two pieces of information: the internal data for the Component, the parameters used to generate the

component, including the parent Components involved. Parameter Viewer displays the parameter values used to generate the selected component in the Analysis Tree, while the Viewer pane shows the view of the Component data, based on the component category. For example, an instance of a TRE Identification Results would contain the TRANSFAC server used, the MATCH parameters, as well as the identifier of the Promoter Sequences Component that was used as input to the MATCH program. Almost all the Components have a textual representation of their internal data that can be written out as a file. These filenames for these are generated based on the name of the Component category, parameters used to generate that component, and the parent Component, if any. These files are typically generated when the Component is created.

Working Directory: When creating a New Analysis, the user is prompted for an Analysis Name, and Working Directory. PAINT then creates a directory by the name of the user specified Analysis Name, in the given Working Directory. The purpose of the Working Directory is to store all the Component data in files, and make the results available outside the PAINT GUI. The annotated screen shot below shows how these concepts correspond to the GUI (Figure 13).

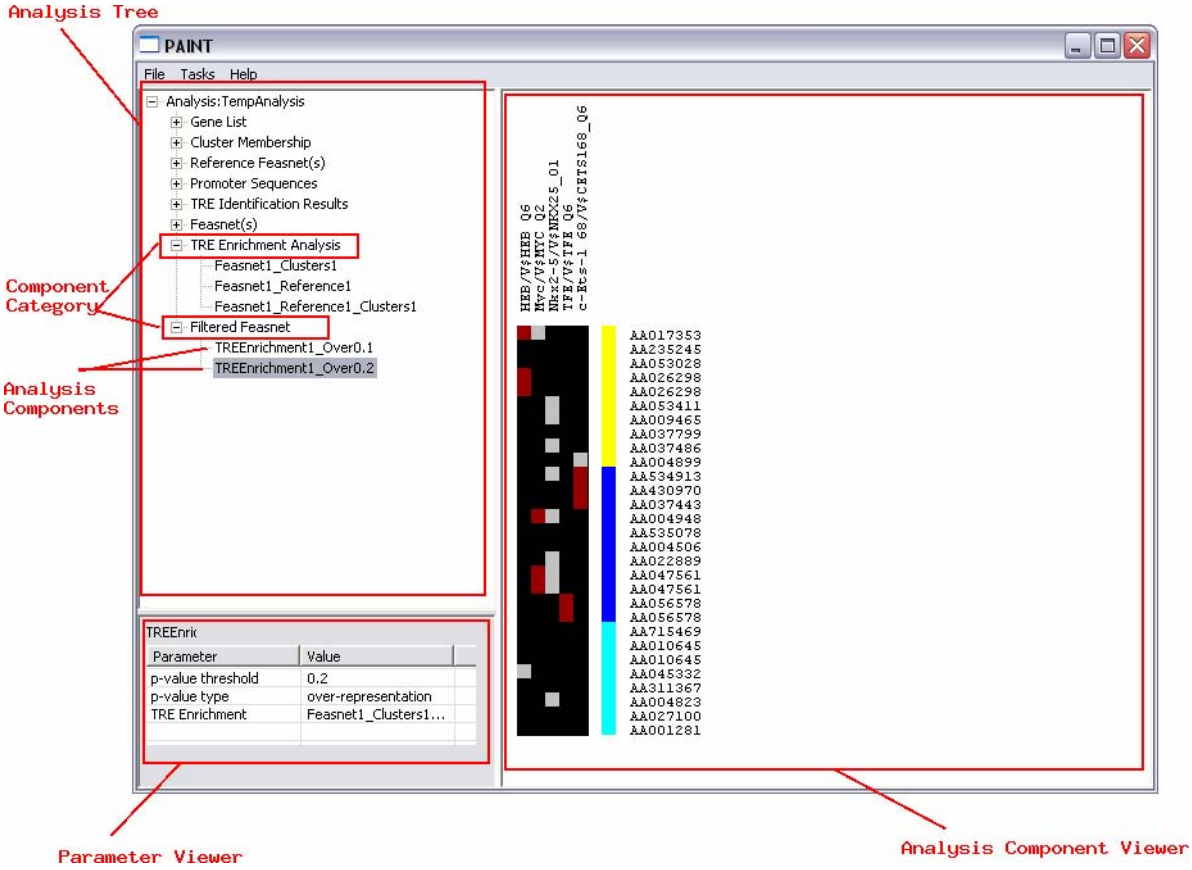

Figure 13: PAINT Architecture Overview

PAINT Components

PAINT Analysis is comprised of eight Components, and eight corresponding Tasks that generate the components. Figure 14 shows the available Tasks from the PAINT UI, a schematic representation of the PAINT Analysis, and the relationship among the components. The ovals are

the components, and the connecting arrows indicate the Task names corresponding to the PAINT UI.

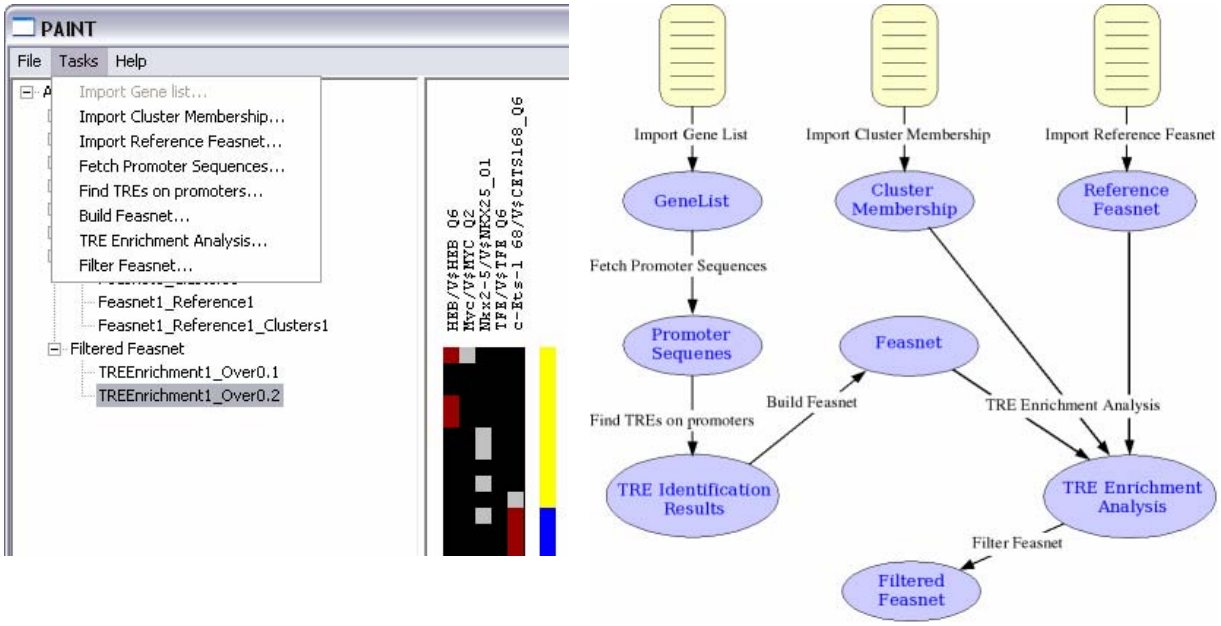

Figure 14: PAINT Tasks in the GUI and the corresponding workflow schematic.

GeneList (Task: Import Gene List)

GeneList contains a list of gene identifiers. Currently supports Accession Number, LocusLink, Ensembl Gene ID, and Gene Symbol. On the BioSPICE Dashboard, the input gene list is obtained from the workflow, by reading in a TimeSeries data type. For more details on the Timeseries format, please refer to the TimeSeries Specification on the BioSPICE web site (BioSPICE account required). Values in the first "column" in the time series, are assumed to be the gene identifiers. Each analysis can have only one Gene List component, since Gene List characterizes an Analysis. In the parameter dialog box shown (Figure 14), when the user selects a file, the first few gene identifiers are shown in the preview box to enable the user to select the right kind of supported gene identifier. Clicking on the OK button creates a GeneList Component that is added in the appropriate component category. When the component is created, the imported gene identifiers are stored in a file in the Working Directory. On the Dashboard, the Task of Import Gene List is executed implicitly, as the TimeSeries input is provided from the Dashboard Workflow.

Cluster Membership (Task: Import Cluster Membership)

This component contains a mapping between the gene identifier from the gene list, and a Cluster ID. In the TRE Enrichment Analysis Task, TREs on the gene promoter sequences of each cluster can be compared to the Feasnet, or the Reference Feasnet. For more details, refer to the TRE Enrichment Analysis description. Typical cluster identifiers could be numbers, or alphabets. The included parameter dialog for Cluster Membership is capable of importing Cluster Membership component from a simple text file. It is possible to have multiple components in the same analyses, as it makes sense to have clustering analyses with different parameters on the same data set. No files are created in the Working Directory for this Component. The expected file format for import is straight forward: Gene identifier, followed by a tab character, and the cluster identifier per each line of a text file. No headers, comment lines or empty rows are allowed. It is

recommended to have the cluster identifiers as short as possible - for example: 1, 2, 3, 4, 5 or A, B, C, D, E etc.

Reference Feasnet (Task: Import Reference Feasnet)
A reference set is a list of genes, of which the input gene list is a subset. A Reference Feasnet is a Feasnet component generated from a selected reference list of genes. In the TRE Enrichment Analysis Task, an input Feasnet is compared to the Reference Feasnet, to compute the significance of the TREs in the input gene list. The selection of appropriate reference set is the key to derive meaningful hypotheses. Comparison of the experiment Feasnet to the entire genome gives erroneous results of the input gene list is obtained from a microarray that does not span the entire genome or is specific to a particular tissue/disease. However, the choice of reference does not end with using the Reference Feasnet from the microarray gene list. For example, in comparison of an early up-regulated gene set to the set of all up-regulated genes, the significantly enriched TREs point to those that are characteristic of early up-regulated genes relative to all the up-regulated genes.

These following steps required to create a Reference Feasnet Component:
1. Choose an appropriate reference as described above.
2. Create a New PAINT Analysis - say, the Analysis Name is Reference and the Working Directory is C:\paint
3. Fetch Promoter Sequences with the desired number of base pairs.
4. Find TREs on the promoters with desired parameters.
5. Build a Feasnet with the desired parameters.
6. Find the file(s) with the extension .fnet in the directory: C:\paint\Reference. The file names have names starting with TREn where n is a number - the order in which the Feasnets appear in the Feasnet component category in the PAINT GUI.
7. Create a new analysis, with the input gene list, and use the desired reference files when importing a Reference Feasnet.

It is important that the desired parameters must match the parameters intended for the gene list of interest. No files are created in the Working Directory for this Component.

Promoter Sequences (Task: Fetch Promoter Sequences)
PAINT has a back end database component that stores promoter sequences of the supported organisms: Human, Mouse and Rat. These sequences are up to 5000 base pairs in length, upstream of the Transcription Start Site, as annotated by Ensembl. This Task is responsible for fetching the promoter sequences from PAINT database. Since it is required to contact the URL shown in the parameter dialog, it is required to have a network access to run this Task. It is possible to have multiple Promoter Sequences components in the same analysis, differentiated by the promoter length.

The only user editable parameter here is the length of the promoter sequences. There is no established rule to select the number of promoter sequences. Typically, 2000 is a good number; if the TRE identification results do not yield expected known TREs, for example, greater promoter sequence length may be required. When the promoter sequences are fetched, they are written to a file (name starts with Promoters) in the Working Directory of the Analysis.

TRE Identification Results (Task: Find TREs on promoters)
PAINT uses web-based MATCH program that finds TREs on the given promoter sequences, based on the TRANSFAC database. Currently, PAINT can connect to TRANSFAC Public server (free registration required), TRANSFAC Pro server (paid license required), or internal TRANSFAC Pro installation (paid license required). Since this requires working with an external web server, Internet access is required. Registration details of TRANSFAC Public can be found at: http://www.gene-regulation.com/register, and more information on TRANSFAC Pro at: http://www.biobase.de/pages/products/transfac.html
The Promoter Sequences drop down list allows the user to select a set of Promoter Sequences on which this Task is executed. In the figure, the selection GeneList1_2000 means the promoter sequences fetched using the GeneList1 component, with the number of base pairs as 2000. The Filter option refers to the MATCH profile for cut-off scores in finding the TREs. Minimize False Positives being the most stringent, followed by Sum of MinFP and MinFN and Minimize False Negatives. Typically, Minimize FP is a good cut off if the downstream analysis does not include other ways to prune the results (TRE Enrichment Analysis does not prune the results, but only assigns p-values to the TREs). This Task takes a while to finish, depending on the number of sequences, as well as the number of promoter sequence length. When the results are fetched, they're stored as HTML files in the Working Directory. The results include the TRE, the TF that can bind to the TRE, the position of the TRE, the strand it is found on, and the matching scores. It is possible to have multiple instances of this component with varying parameters.

Feasnet (Task: Build Feasnet)
This Task enables further filtering of the TRE Identification Results. The first filter, Core Similarity Threshold can be adjusted between 0.9, 0.95, and 1.0, 1.0 being the perfect match of the potential binding site on the input promoter sequences with the TRANSFAC database. It is yet unclear whether to include the TREs found the complementary side of the promoter sequence. Though it is known that some TREs on the complementary strand do play a role, looking at all the matching sites on the complementary strand may mean more false positives. This Task makes it possible to build multiple Feasnets from the same TRE Identification results, but varying the filters. That way, the user can look at the Feasnet to make a decision on which Feasnet(s) to use for further analysis. Each generated Feasnet is written in the Working Directory, with the pattern: TREIDn_coresim_comp.fnet, where TREIDn is the TRE Identification Component used, coresim is the core similarity filter that takes one of the {0.9, 0.95, 1.0} values, and comp indicates whether the TREs on complementary strand are included.

TRE Enrichment Analysis (Task: TRE Enrichment Analysis)
This is one of the key steps in PAINT analysis. Before executing this Task, it is required to import the desired Reference Feasnet and Cluster Membership components, which then appear in the drop down lists shown in the parameter dialog. The three kinds of comparisons possible are: Feasnet to Reference Feasnet, Cluster to Feasnet and Cluster to Reference Feasnet. The first analysis implies how the TREs in the selected gene list compare to the Reference Feasnet. The interpretation depends on how the gene list was selected, and the selected Reference. For example, if the gene list comprises all the differentially expressed genes (relative to a control), and the reference is the list of all genes on the microarray, the interpretation could be that the highly enriched TREs are responsible for the differential expression. The latter comparisons deal with the cluster membership, which typically is obtained from a gene expression pattern. In these cases, enriched TREs found on the genes in a given cluster to the reference can be attributed to the observed gene expression pattern, depending on how distinct the pattern is. The hint text below the options (as seen in the Figure 3g, each cluster subset compared to the Reference

Feasnet) indicates the kind of enrichment analysis chosen. For the selected Reference and Cluster Membership, p-values of the TREs are computed using Fisher's Exact Test. The resulting p-values are stored as tab-delimited text files in the Working Directory.

Filtered Feasnet (Task: Filter Feasnet)
This is the output component of PAINT analysis - a hypotheses Gene-TRE interaction network. The "connections" (or interactions) in the network are based on filtering the TRE Enrichment Analysis results. The parameter dialog allows two basic filters, both acting on the statistical significance of the TREs. It is possible to disable the at the most filter or both the filters to look at the TREs. In the Component Viewer, the presence of a TRE (column) on the gene promoter sequence (row) is indicated by a colored square. A red color indicates over-representation or enrichment, and a cyan represents under-representation, the brightness of the color indicating the significance of the TRE. A gray square means that it is statistically not possible to imply that the TRE is significant in the list. If Cluster to Reference Feasnet, or Cluster to Feasnet was chosen in the TRE Enrichment Analysis, a cluster bar appears next to the gene identifiers that indicates that cluster the gene belongs to. The result of this component is a subset of the Feasnet, and is written as a tab-delimited text file in the Working Directory. It is possible to export this network as a GeneTF datatype for Dashboard integration.

7 Case Studies

7.1 Experimental System for Neuronal Differentiation and Adaptation studies

The model system considered presently is the N1E-115 neuroblastoma cell line (Amano et al., 1972). After differentiation, N1E-115 cells synthesize several neurotransmitters and express functionally coupled neurotransmitter/neuropeptide receptors (Richelson, 1990). For use with N1E-115 cells we are currently printing ~8,600 mouse cDNAs which are greater than 95 per cent nonredundant onto 1-inch x 3-inch glass microarrays. This collection of cDNAs was derived solely from CNS tissues as part of the Brain Molecular Anatomy Project at the University of Iowa (http://brainest.eng.uiowa.edu/index.html). The set represents ~3,200 annotated genes and ~5,400 unannotated genes using conservative definitions of annotation.

The data for Case Study 1 was obtained by comparing gene expression in undifferentiated growing N1E-115 cells to that in differentiated N1E-115 cells. A set of 193 annotated genes that are differentially expressed at least by a factor of 2 was considered for further analysis.

For the Case Study 2, the dataset is obtained from microarray experiments of differentiated N1E-115 cells exposed to the neuropeptide angiotensin II (AngII). Ang II is a multifunctional hormone that influences the function of cardiovascular cells through a complex set of intracellular signaling pathways initiated by the interaction of Ang II with the AT1 and AT2 receptors (Berry, 2001; Touyz, 2002). AT1 receptor activation leads to cell growth, vascular contraction, inflammatory responses and salt and water retention, whereas AT2 receptors induce apoptosis, vasodilatation and natriuresis. In an effort to isolate the transcriptional response to AngII to the AT1 receptor (AT1R), the AT2 subtype was blocked with a saturating dose of antagonist. Cultures of differentiated N1E-115 cells were pretreated with 10uM PD123319 for 30 min by addition of a 1000X stock solution directly to the culture media without removal of the dish from the incubator. After 30min, AngII (100nM final, 100uM stock) was rapidly added to all the parallel cultures required for the time course. Pretreated cultures with no AngII added were considered as time=0 samples. A time series of gene expression data was obtained from microarray experiments with RNA isolated at 0, 5, 15, 30, and 60 minutes after exposure to AngII. A total of 1338 genes with at least twofold change at any of the time points were considered responsive and were included in further analysis with PAINT version 2.3 (Note that the current PAINT version as of January 2006 is 3.3).

PAINT 2.3 contains sequences from version 7.3a of the Ensembl annotated mouse genome database. This mouse draft sequence is based principally on whole genome shotgun sequencing of around 7x coverage. This was frozen in February 2002 and incorporates finished clone information where available. The sequence is estimated to cover 96 percent of mouse euchromatic DNA. A total of 22,444 genes that are annotated were processed of which 21,363 promoter sequences were retrieved based on the TSS identification and filtering criteria specified in the Methods section.

7.2 Case Study 1: Neuronal Differentiation

As described above, an example microarray data set was obtained by comparing growing and differentiated neuroblastoma N1E-115 cells in culture providing 193 annotated genes that were differentially regulated at least by a factor of two (130 up-regulated and 63 down-regulated). The GenBank accession numbers of these 193 clones were provided as the input to PAINT.

The Upstreamer module returned a list of 155 promoters, *i.e.*, 38 of the 193 genes of interest did not have a promoter sequence that met the filtering criteria (specified in Methods) for

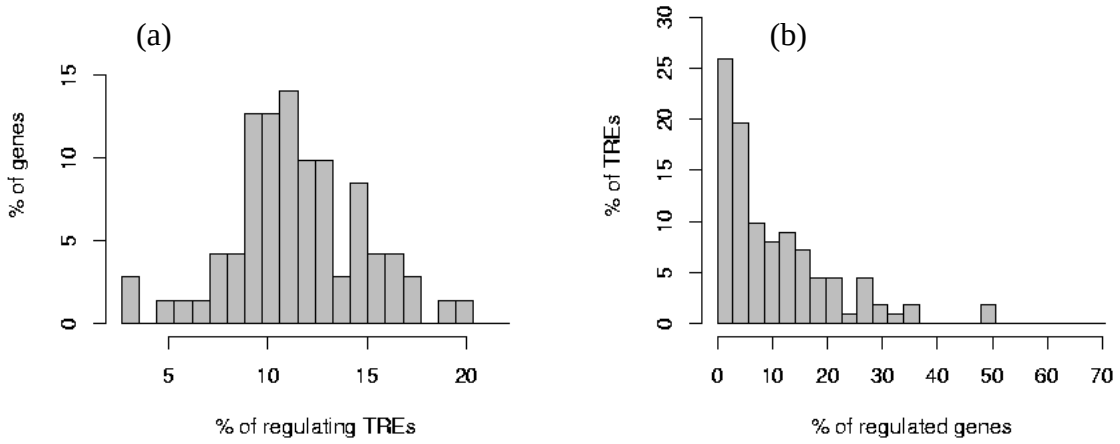

Figure 15: The distribution of interactions for (a) the genes and (b) the TREs in the DIFF155 interaction matrix. There are relatively few genes that are regulated by either too large or too few number of TFs, indicating relatively rare cases of excessive or little regulation. The interaction distribution for transcription factors indicates that there are relatively few transcription factors that can regulate majority of the genes in the input list.

constructing the UpstreamDB database. This data set is referred to as DIFF155 for the rest of the document. Of the 155 promoters retrieved, 107 correspond to the up-regulated genes (referred to as DIFFUP107) and 48 to the down-regulated genes (referred to as DIFFDOWN48). These 155 promoter sequences were processed using the TFRetriever module communicating with the MatInspector software. Within MatInspector, the vertebrate database with 128 TRE families and 313 position weight matrices was utilized in motif inspection. The TFRetriever module retrieved motif matches for a total of 273 distinct TREs. FeasNetBuilder constructed a candidate interaction matrix between the 155 genes and 273 TREs. The CIMs for the DIFFUP107 and DIFFDOWN48 subsets were obtained from the DIFF155 CIM.

The distributions of interactions for the 155 genes and the 273 TREs are depicted in Figure 15. The distribution is unimodal with long tails (low percentage of genes at the extreme ends of the distribution). This indicates that relatively few genes are regulated by very many or very few TFs (extreme ends of distribution shown in Figure 15a). The distribution shown in Figure 15b indicates that a few TREs are present on significant fraction of the genes of interest indicating potential role in system-wide effects (right end of the distribution). Analysis of Figure 15b also indicates that most of the TREs are present on and hence can regulate relatively few genes each (suggesting a function in fine-tuning, local effects).

A representation of the candidate interaction network is depicted in Figure 16. A subset of the CIM is shown in Figure 17. The genes and motifs were individually clustered using binary distance as the dissimilarity metric. The column immediately next to the CIM represents whether the corresponding gene is found to be up-regulated or down-regulated in the expression data. It should be noted that most of the up-regulated or down-regulated genes do cluster together based on the regulatory pattern of their promoters. However, there are clusters containing both up and down regulated genes indicating that the activity of specific transcription factors in the experiment needs to be utilized to prune the candidate interaction matrix to improve the network

prediction. A network layout diagram containing five TREs in DIFF155 CIM with the lowest p-values for over representation is depicted in Figure 18.

The p-value of each TRE in CIMs for DIFF155, DIFFUP107 and DIFFDOWN48 was calculated using the StatFilter submodule. Analysis of the TREs that are significantly over represented in DIFF155, DIFFUP107 and DIFFDOWN48 revealed details that are very different from that of the analysis of the static *cis*-regulatory pattern (complete CIM). A total of 42 TREs were significantly overrepresented in at least one of DIFF155, DIFFUP107 and DIFFDOWN48. A p-value threshold of 0.1 for DIFFUP107 and DIFFDOWN48, and of 0.15 for DIFF155 was employed to filter for TRE significance. It is interesting to note that only two families of TREs (MYOD and FKHD) were significantly over represented in both the up-regulated (DIFFUP107) and down-regulated (DIFFDOWN48) genes. Such a dramatic difference was not obvious from the analysis of the feasible cis-regulatory pattern based on all the TREs found to be present (Figure 16). Several of the TRE families are implicated in cell differentiation and maturation (for example, AREB: Ikeda et al., 1995; CREB: Dobi et al., 1995; GATA: Nardelli et al., 1999).

Note that the results from the analysis of significantly over-represented or under-represented TREs may be a conservative estimate of the TREs involved in regulation in the experimental context of interest. In the present case study of N1E-115 differentiation, some TREs that are known to be involved in the differentiation of other cell types were not found to be significantly

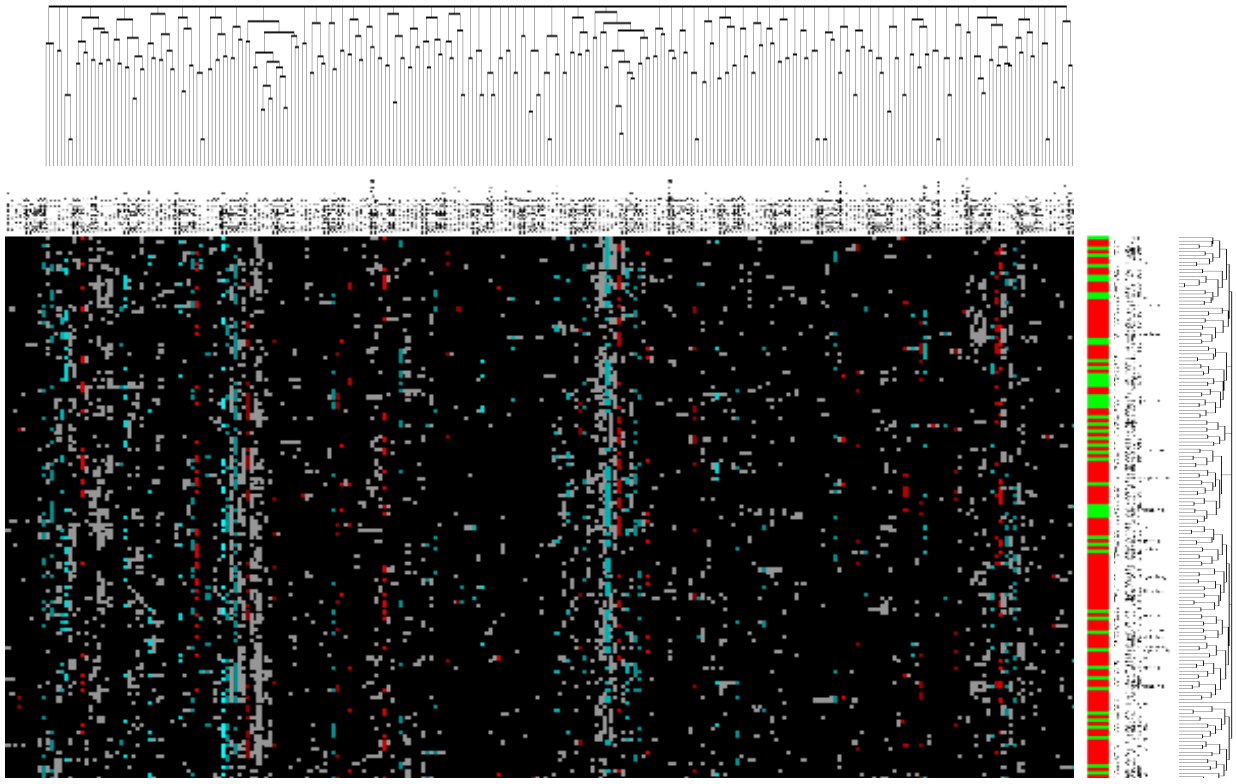

Figure 16: Representation of the candidate interaction matrix for 155 differentially expressed genes and 273 TREs analyzed in N1E-115 cell differentiation. Dataset is clustered using binary distance as the dissimilarity metric. Individual elements of the matrix are colored by significance p-values: over representation in the matrix (indicated in red), under representation (indicated in cyan), and the TREs that are neither significantly over nor under represented (colored in gray). The column to the left of the matrix represents whether the gene expression in differentiated cells: red indicates up-regulated genes and green indicates down-regulated genes.

over-represented. One example is MZF1, not found to be significantly enriched in either of the up- regulated or down-regulated genes (p-value of 0.77 in DIFF107 and p-value of 0.59 in DIFF48), even though it has been shown to be involved in delaying cell differentiation in other cell types (Morris et al., 1994). MZF1 and similar TREs may have not appeared to be significant in the present case study because they are not involved in N1E-115 differentiation, or because the data currently available was not sufficient to identify every TRE that is involved in the process.

Another interesting observation was that it is possible that a particular TRE is not found to be significantly over-represented or under-represented in a particular cluster/subgroup of genes, but can still be significantly over-represented in the overall set of genes considered: NFAT, LEFF, and GATA/GATA3.01. These are implicated in cellular differentiation in different neuronal cell types (NFAT: Plyte et al., 2001; GATA/GATA3.01: Nardelli et al., 1999).

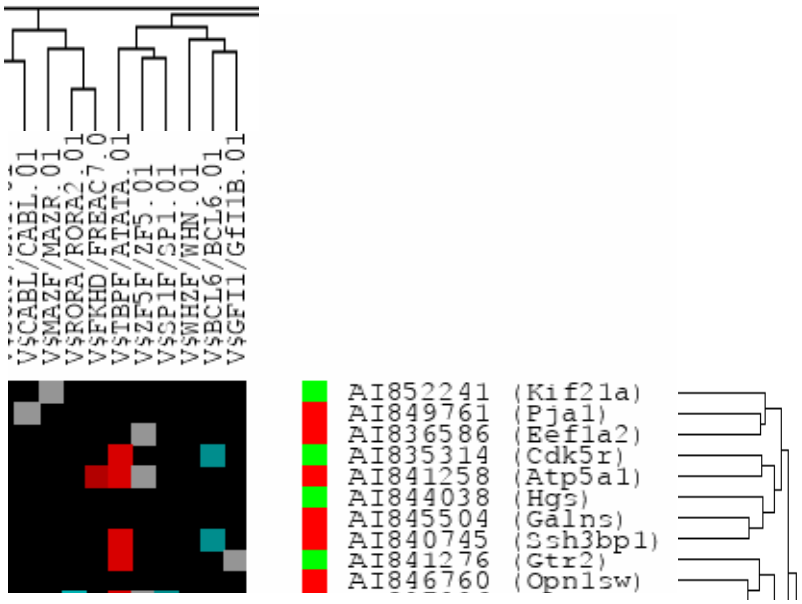

Figure 17: A subset of the candidate interaction matrix for DIFF155.

Given no other information, an identification algorithm would have to compute 71*87=6177 connection parameters. However, since the candidate network contains 883 nonzero entries, *i.e.*, only 883 (14.2 per cent) interaction parameters need to be computed. Even this is a gross overestimate as the dynamic activity data about specific TFs from ChIP experiments can substantially reduce the number of candidate interactions further. The localization data thus obtained can significantly improve the regulatory network identification (Zak et al., 2001; Hartemink et al., 2002).

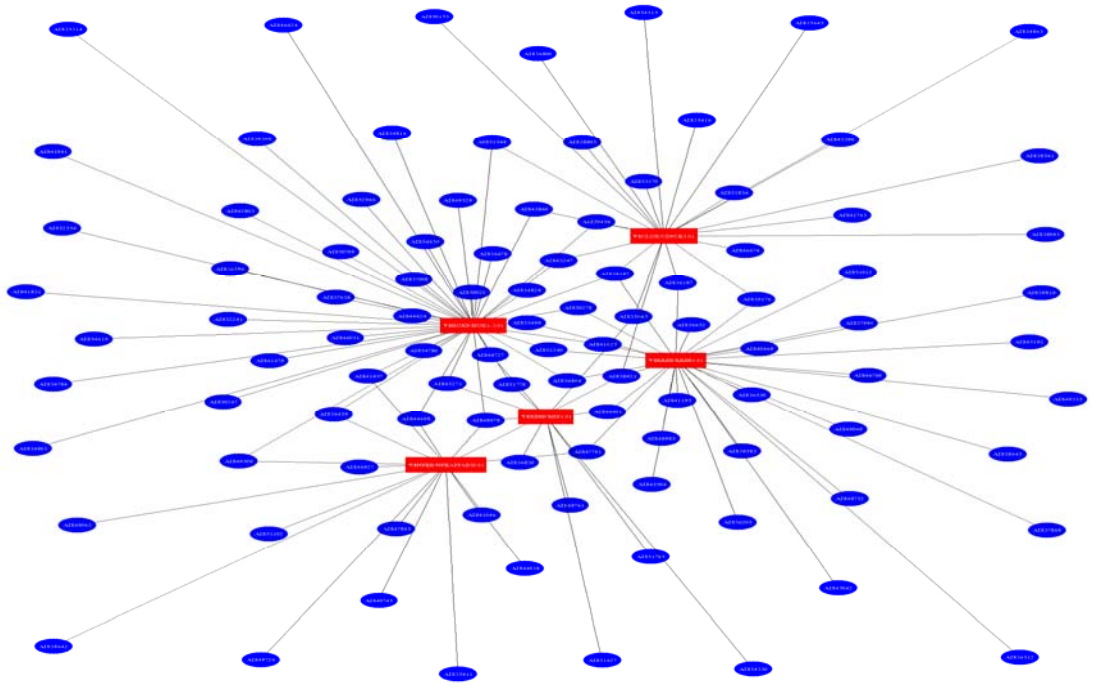

Figure 18: A network layout showing the top five TREs in DIFF155 CIM ($p<0.1$). The rectangular boxes represent TREs and ellipses represent genes.

7.3 Case Study 2: Neuronal Adaptation

As described above, a gene expression time series data set was obtained from microarray experiments involving neuroblastoma N1E-115 cells at {0, 5, 15, 30, 120} minutes after exposure to angiotensin II. Of the ~8600 genes on the microarray, a total of 1338 genes with at least two-fold change at any of the time points were considered responsive and were included in the analysis. This list of 1338 genes was presented as input to PAINT for promoter analysis.

The PAINT Upstreamer module retrieved 578 promoters (referred to as ANG578 for the rest of

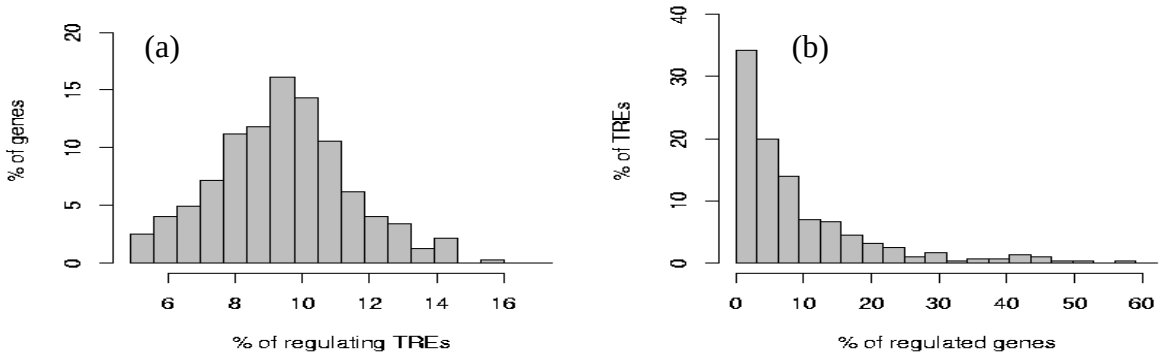

Figure 19: The distribution of interactions for (a) the genes and (b) the TREs in ANG578 CIM data. There are relatively few genes that are regulated by either too large or too few number of TFs, indicating relatively rare cases of excessive or little regulation. The interaction distribution for TFs indicates that there are relatively few TFs that can regulate majority of the genes in the input list.

the document). The promoters for almost all the annotated genes in the initial set of 1338 genes were retrieved. The connectivity of the ANG578 CIM is depicted in Figure 19 above. The connectivity in CIM for ANG578 was qualitatively similar to that observed in the DIFF155 data set (case study 1). As in Case Study 1 there were few genes that could be regulated by very many or very few TFs (Figure 19a). Again, as in Case Study 1, analysis of the distribution shown in Figure 19b indicates that relatively few TFs can regulate large number of genes in ANG578 (right end of the distribution: system-wide effects) and most of the TFs regulate few genes each (left end of the distribution: fine-tuning, local effects).

A representation of the candidate interaction network is shown in Figure 20. The genes and TREs are individually clustered using binary distance as the dissimilarity metric. Each row in the column immediately next to the dendrogram represents the cluster number of each gene from clustering the expression data. Note that many of the genes in an expression-based cluster also cluster together based on the regulatory pattern of their promoters. However, this analysis is based on the static structure of the CIM. The activity of specific TFs in the experiment needs to be utilized to prune the candidate interaction matrix for improving the network prediction.

The p-value of each TRE is calculated by the StatFilter module. Based on a p-value threshold of 0.1, several TREs were shown to be over represented in the ANG578 CIM network. Several of the factors binding to these TREs are known to be involved in cellular response to angiotensin II: AP1F and EGRF - Lebrun et al., 1995; CREB - Cammarota et al., 2001; NFKB - Wolf et al., 2002. Also several TREs that are known to play role in neuronal development are also over represented in the ANG578 CIM: RBPJK - de la Pompa et al., 1997; HEN1 - Bao et al., 2000; LMO2COM - Yamada et al., 2002.

The transcription factor family STAT is shown to be involved in response to stimulation of AT1R (Mascareno & Siddiqui, 2000). However, the TREs corresponding to the factor STAT were not found to be over represented in ANG578 (p-values of all TREs binding to STAT family of TFs was close to 0.5 in ANG578). Further PAINT-based analysis of individual clusters of genes in ANG578 may provide the group(s) of genes with over represented TREs for the STAT family of TFs, however.

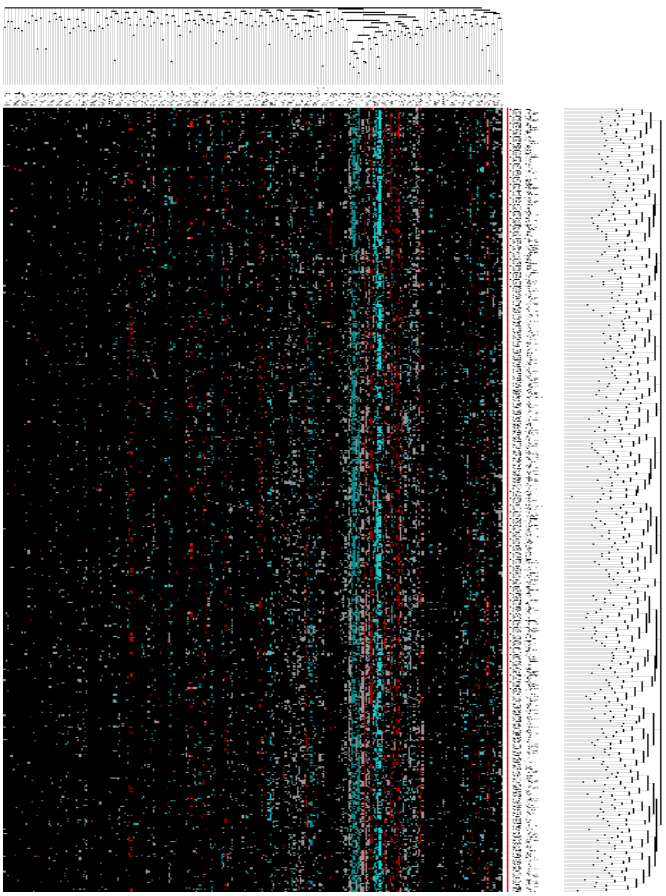

Figure 20: A representation of the candidate interaction matrix for 578 differentially expressed genes and 291 TREs analyzed in ANG578. The dataset is clustered using binary distance as the dissimilarity metric.

7.4 Case Study 3: Circadian Rhythms

We have been focusing on analyzing the microarray gene expression data in the SupraChiasmastic Nucleus (SCN) region of the brain. SCN is a well studied system, and is ideal for the study of circadian rhythm, as the gene expression shows a clear clock based cycling of genes of interest, and outputs important to physiology: protein levels and spiking activity. Effects of Modulatory inputs such as EGF in the SCN can also be studied. This offers a good opportunity towards integrating the gene regulation with the spiking physiological output to create true whole cell input-output models. In other words, the question we are asking is: **How do changes in gene activity cause changes in the neuron activity/function?**

In this Usecase, our objective has been to understand the gene regulatory network, and the key regulators that drive the free running biological clock. In the future, this knowledge can be extended to understand the effects of modulators of the clock, such as Epidermal Growth Factor (EGF). We have been working with microarray gene expression data from this publication (Figure 21): Coordinated transcription of key pathways in the mouse by the circadian clock, Cell 109(3):307-320, 2002. The gene list of interest is the set of differentially expressed genes in the SupraChiasmastic Nucleus (SCN) from the clusters CT6, CT10, and CT18 (148 genes out of a total of 292 genes for which promoters exist in PAINT database). The question we seek to answer here is: "What are the candidate regulatory networks characteristic to clusters CT6, CT10 and CT18 relative to all the differentially expressed genes?"

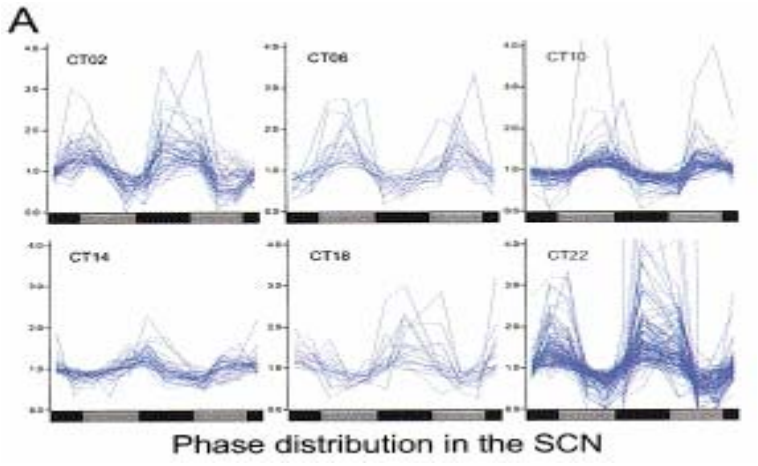

Figure 21: Figure 1A from Panda et al., Cell, 2002. Circadian Regulated Genes in the SCN. 337 SCN transcripts were determined to be circadianly regulated by COSOPT, and they were binned into six circadian phases, 2, 6, 10, 14, 18, and 22, by cluster analysis. Data traces of different Phase cluster in the SCN are shown. Values in the x axis represent hours after the first subjective dawn.

Our analysis focused on the TREs on the promoter regions of the genes of interest, to identify the significant TREs, as well as the TFs that can bind to these TREs. A hypothesis network was generated based on significantly enriched TREs from the clusters CT6, CT10, CT18, relative to all the differentially expressed genes Of these, the significance scores of the binding site for AP-1 was found to be in agreement with the binding activity of the Transcription Factor AP-1, as illustrated in the Figure 22. (Francois-bellan et al., Brain Res Mol Brain Res. 2000 Dec). Figure 4 also shows the significance scores of the TREs seen in the Figure 3, as well as the TRNA workflow on the Dashboard 5.0.

We have started looking at the combinatorial effects of the TREs. Combinatorial regulation addressed here is based on the concept of Composite Elements (CE). Composite regulatory elements contain two closely situated binding sites for two distinct transcription factors. Specific factor-DNA and factor-factor interactions contribute to the function of CEs. Cooperative action of TFs within the CEs result in highly specific pattern of transcription, which cannot be provided by involved factors separately. Current efforts focus on analysis of promoter regions of co-expressed genes, to find CEs and generating interaction networks of these genes.

We have explored the dependence of the hypotheses on changes in the algorithmic parameters to identify robust hypotheses on CREs. In this case, we varied the similarity score in matching to known CREs from 0.7 to 1.0 in steps of 0.05, and considered those with 0 or 1 base pair mismatch, yielding a total of 7*2=14 parameter choices.

Only those CREs that were enriched for all parameter variations were considered in the hypotheses.

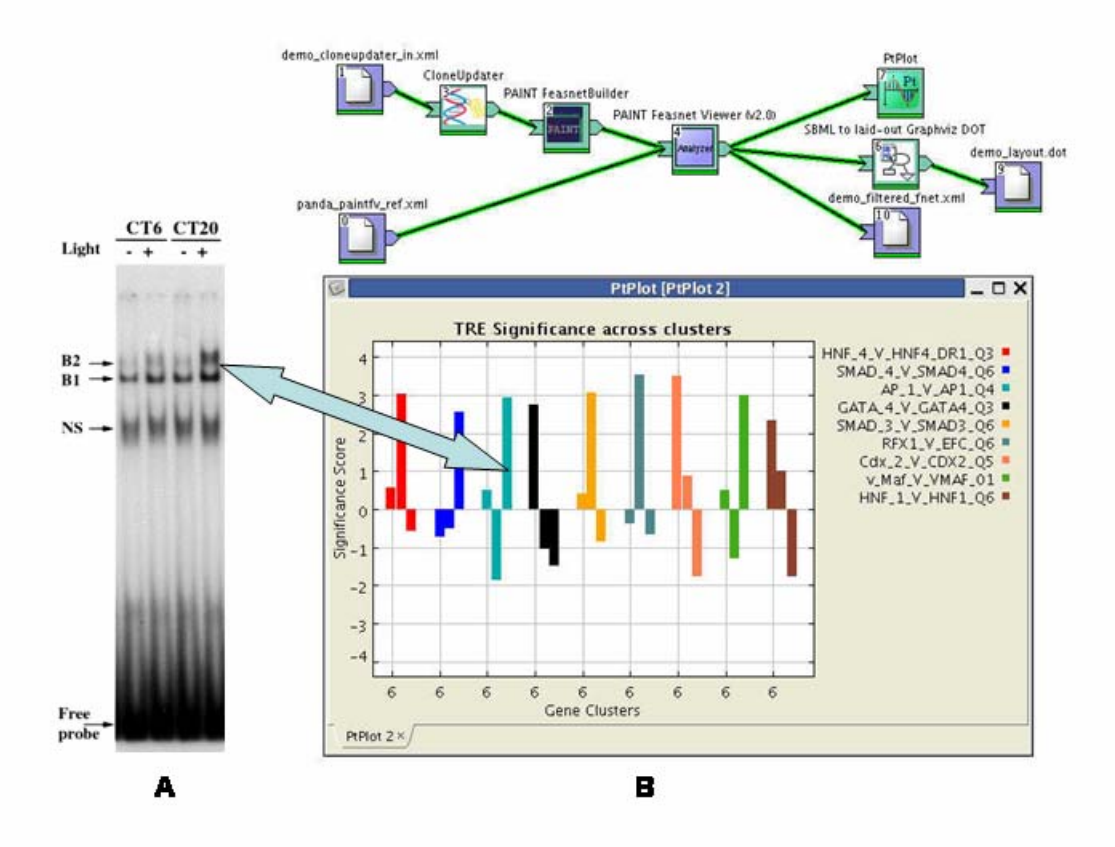

Figure 22: Experimental data for AP-1 activity corresponding to TRE significance scores

Based on the results, we have made robust hypotheses on functionally related TRE pairs (CREs) relevant to the gene regulation in free-running circadian clock neurons. The results are shown in Figure 23 as significance scores, calculated as -log (p-value) to allow us to focus on order of magnitude differences in enrichment. One of the interesting results we came across from this analysis is that, the TRE AP-1 binding site from TRE Enrichment Analysis is enriched in the CE Enrichment Analysis as well: AP-1/CEBPbeta3. Our results extend the experimental findings

with the hypothesis that the observed increased activity of AP-1 is functioning as part of combinatorial regulation with CEBPbeta3.

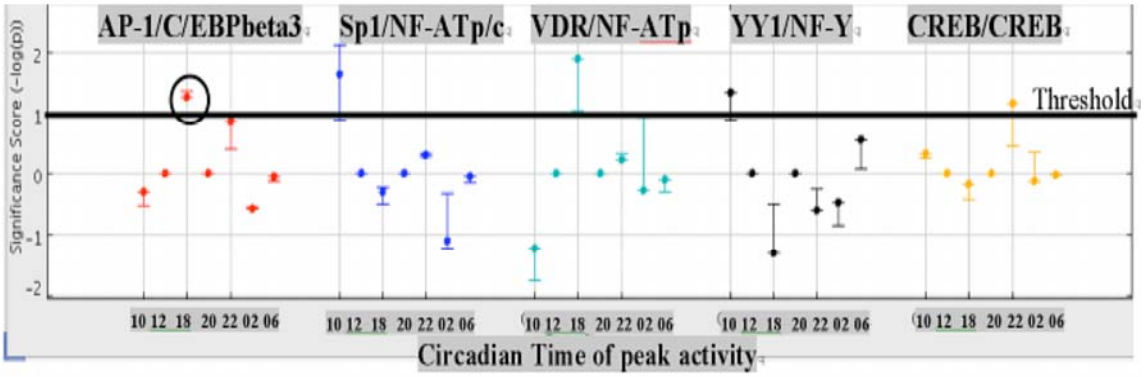

Figure 23: Significance scores of Composite Elements. Circled is the CE AP-1/CEBPbeta3

7.5 Case Study 4: Pre-apoptosis in Kidney cells exposed to the pathogen Staphylococcal Enterotoxin B (SEB)

TRNA as applied to gene expression data from SEB-induced apoptosis, in the Human Renal Proximal Tubule Epithelial Cells (RPTEC). The primary objective of this analysis is to augment an existing model based on contextual biological knowledge, with key TFs, potential regulation of key genes by these TFs, and a regulatory network of these TFs. Our analysis identified 18 significant TFs, 7 of which are already present in the aforementioned model, and the rest are found to be potentially relevant, and will be added to the model.

In this case study, the data constituted of: a) Gene expression timeseries of about 6700 genes from the microarray array experiment of the RPTEC exposed to 50microgram of SEB, measured at 2, 4, 6, 8, 12 and 24 hour timepoints after the exposure, (referred to as RPTEC6699 henceforth), b) List of 909 differentially expressed genes from RPTEC6699, with log ratios of expression values (DIFF909) and c) List of 115 genes and TFs present in the model (MODEL115, TFs respectively) . Our objectives in this analysis were: a) Identify the statistically significant TFs that may be correlated with the gene expressed under study, b) find significant TF-gene interactions, and c) Hypothesize a regulatory network of the significant TFs. Our analysis and results are presented in the following three sections, each section corresponding to one of the objectives listed.

TRE enrichment analysis

The purpose of this phase is to identify statistically significant TREs, and thereby the TFs that can bind to these TREs, in the context of the given gene expression. It is also required to rank the TFs based on the significance that allows the downstream analyses, such as TF binding activity prediction, to pick as many TFs as dictated by computational and other practical constraints. Our approach was to select an initial set of significant TFs based on TRE enrichment, and rank them based on cluster specificity score of the selected TFs. The details of the analysis methods are described below. We used the Multi-experiment Viewer program from TIGR (www.tigr.org) for clustering the DIFF909 dataset, to relate the gene expression pattern to transcriptional regulation. We considered only those genes that have valid expression values for all the given timepoints, the amount being about 525 genes (DIFF525). The results of the cluster analysis are shown in Figure 24. Of these, we selected only those clusters that have a distinct expression pattern,

constituting of genes that are clearly co-expressed. Specifically, a list of 405 genes that belong to the clusters: A, B, D, F, G, I, J, K, L, and O are picked for TRE enrichment (referred to as DIFF405 henceforth).

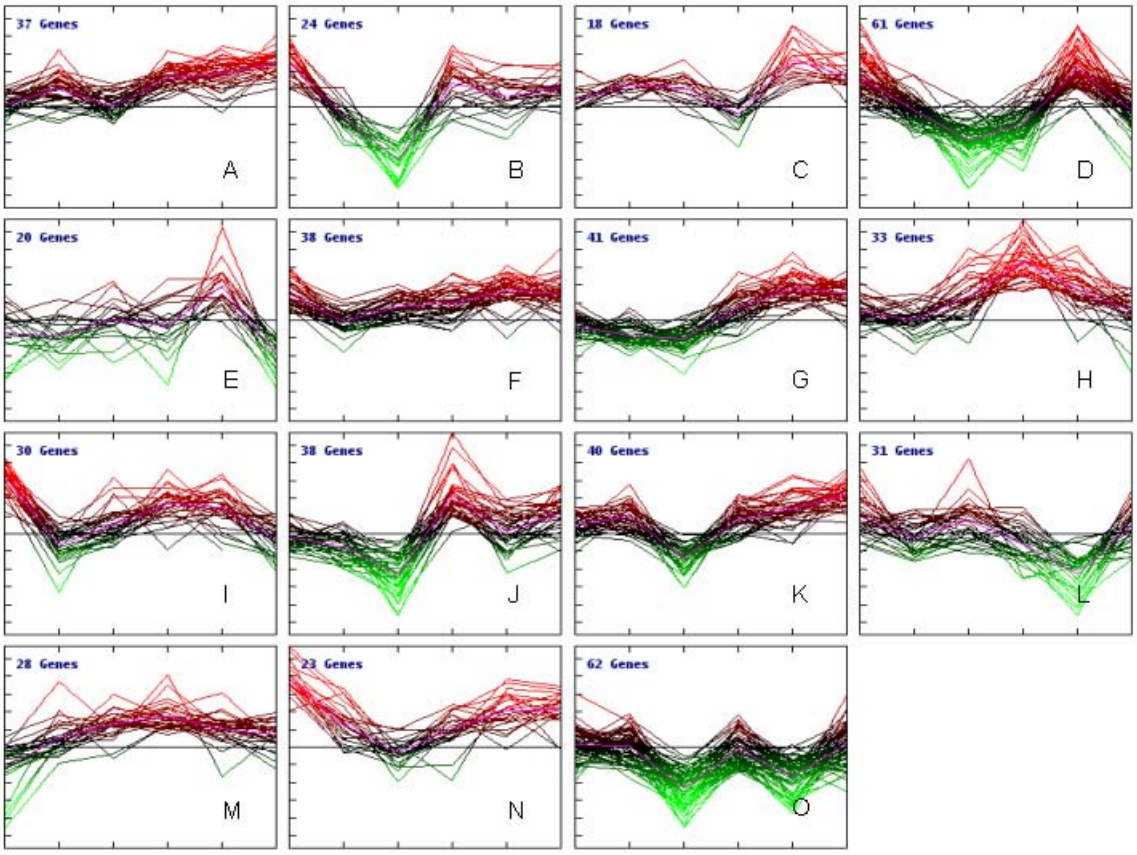

Figure 24: DIFF525 cluster analysis results using k-means algorithm in TIGR MeV software.

For the genes in the DIFF405 set, promoter sequences of up to 5000 basepairs were retrieved using PAINT 3.2 (Vadigepalli et al., http://www.dbi.tju.edu/dbi/tools/paint3.2) Human promoter database. TREs and the corresponding TFs on these promoter sequences were identified using TRANSFAC® 8.3 database and the corresponding Match® program. The TRE enrichment scores are based on the TREs present on the promoter sequences of the genes that belong to each of the clusters in DIFF405, relative to the two reference sets: DIFF405, and RPTEC6700. The interactions were analyzed for sensitivity of the parameter choices using six different combinations of the parameters, namely, the Core Similarity score, whether to include the TREs on the complementary strand of a promoter sequence, while minimizing the false positives (Match reference) 1) 0.90 Core.sim., (+) strand only, 2) 0.90 Core.sim., (-) strand included, 3) 0.95 Core.sim., (+) strand only, 4) 0.95 Core.sim., (-) strand included, 5) 1.00 Core.sim., (+) strand only, and 6) 1.00 Core.sim., (-) strand included. Statistical significance scores of TRE occurrences were computed based on Fisher's Exact Test (Vadigepalli et al., 2003) across all the considered parameters, and relative to both the reference sets. TREs that passed the set threshold (p-value <= 0.05), in at least half of the considered parameter sets were selected for further analyses. In addition, Co-clustering score for each TRE is calculated based on the shared expression clusters for each of the genes that are targets of given TRE. The score calculated in

this scheme is based on the probability that the target genes of a TRE in multiple clusters are selected as a random sample from the full gene list divided into clusters based on expression data. Co-clustering scores of the TREs were used to rank the selected TREs, with the idea that the more specific a TRE is to a gene expression cluster or clusters, the higher is the priority to include the TF associated with the TRE in the model. Table 2 shows the list of the selected TFs based on the co-clustering rank and the individual TRE enrichment analysis. The list of the TREs thus obtained was subsequently filtered based on the ratio of the number of genes that have a TRE, relative to the total number of genes. The overall TF hypotheses that resulted from combining multiple enrichment analysis results, co-clustering scores, and additional count-based filters are shown in the Table below. Some of these are already present in the existing model and the remaining TFs are the potential hypothesized components that need to be added to the model.

Significant Gene-TF interactions

Some of the TFs that were in the model, believed to be important in the biological context, did not meet our stringent thresholds set for statistical significance. These TFs were added to the list of the selected TFs from Table 2, to be analyzed for potential transcriptional regulation of the genes in the model. In effect, the resulting network structure would consist of regulators based on prior knowledge, as well as TRE enrichment analysis. The MODEL115 gene list was run through PAINT 3.2 and TRANSFAC with the most restrictive of Match parameters (core sim:1, compl strand: no) to limit the results to only high confidence TREs. The resulting candidate interaction matrix of TREs on the genes was filtered for only those TREs than can potentially be bound to, by the selected TFs.

TF Regulatory Network analysis

To enable modeling of the transcriptional regulation, it is essential to "close" the network, i.e., the regulation of the TFs themselves must be considered. Since the genes included in the model are based on known biological function, only the selected TFs are considered to be the significant regulators of the genes encoding the selected TFs. Up to 5000 basepairs of promoter sequences of the encoding genes were retrieved from PAINT 3.2 Human Promoter Database, and run through TRANSFAC Match with most restrictive parameters (Core sim: 1.0, minfp, compl: no). The resulting candidate interaction matrix is filtered for only the selected TFs. The resulting network structure is shown in the Figure 25.

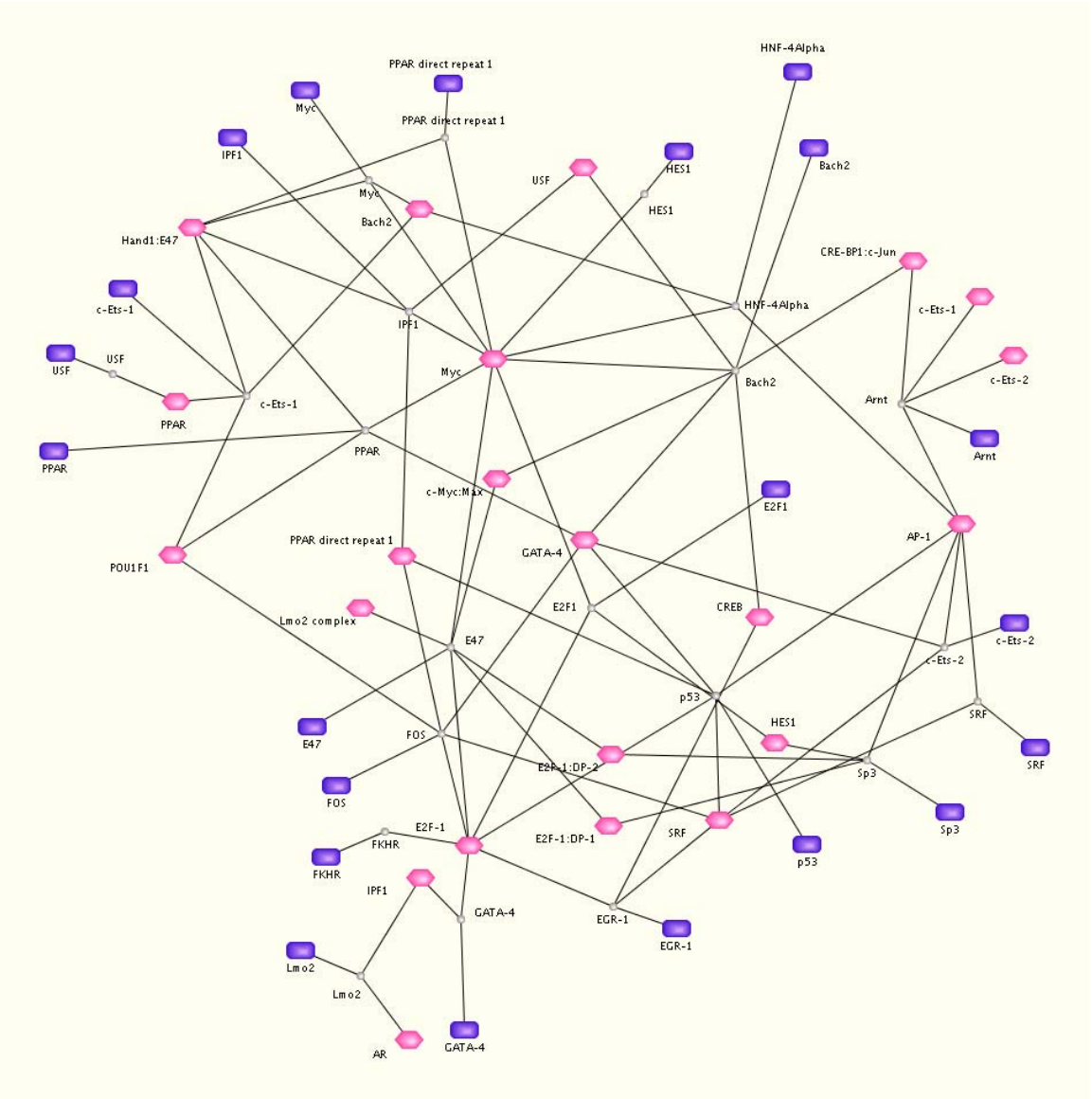

Figure 25: Regulatory network involving the PAINT predicted TFs (see Table 2 below).

Table 2: A summary the PAINT predicted transcription factors relevant to the SEB response in the kidney (RPTEC).

TF name	Biological processes	In model?
E47	Causes acute myeloid leukemia; induces enhanced proliferation; apoptosis	Yes
c-Ets-1	Immune response; negative regulation of cell proliferation	Yes
CRE-BP1:c-Jun	Mediates activation in response to UV and other cellular stresses< /td>	Yes
c-Myc:Max	Cell proliferation control; may induce apoptosis	Yes
SRF	Interacts with several essential TFs	Yes
E2F-1	Regulates Apaf-1, the gene for apoptosis protease-activating factor 1	Yes
NF-KappaB	Key regulator of genes involved in infection, inflammation and stress< /td>	Yes
HNF-4alpha	Blood coagulation; lipid metabolism	Potential
AR	Cell proliferation; differentiation	Potential
Arnt	Adaptive response to hypoxia	Potential
Lmo2 complex	Regulation of red blood cell development	Potential
IPF1	Glucose-dependent regulation of insulin gene transcription	Potential
Bach2	Transcriptional repressor, activator; B-cell specific	Potential
PPAR	Key regulator of adipogenesis	Potential
POU1F1	Negative regulation of cell proliferation	Potential
SMAD-4	May act as a tumor suppressor	Potential
USF	Activates CEBPA that is related to body weight homeostasis, leukemia	Potential

8 A Novel Methodology for Structured Modeling of Gene Regulatory Networks

We now describe a methodology for the structured modeling of gene regulatory networks. It consists of two distinct parts: 1) determination of nuclear connectivity, and 2) model identification. A schematic diagram is shown in Figure 26. Even though both the model (parameter) identification (box in lower right-hand corner) and nuclear connectivity determination make use of the gene expression data directly, they are decoupled, with the nuclear connectivity being fed to the model identification as prior knowledge.

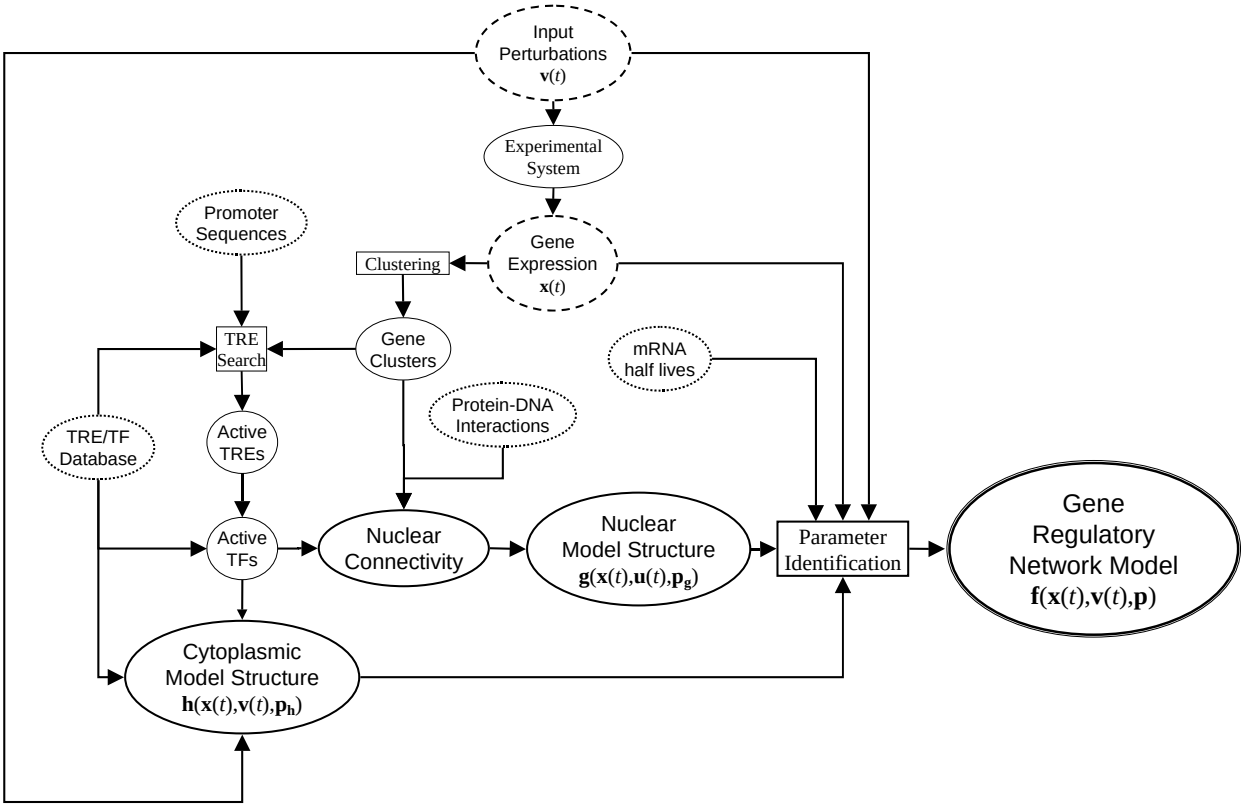

Figure 26: Integrated components of the structured approach to gene regulatory network identification. Information is designated by ovals, actions are designated by rectangles. External sources of information are indicated by dotted ovals and include promoter sequences of all the genes as obtained from genomic sequences; databases describing TREs and the TFs that bind to them; protein-DNA interaction data; and half lives for the transcripts of all of the genes. Internal sources of information that would be collected from a particular experiment are indicated by dashed ovals and include genome-wide gene expression data and specific input perturbations. Actions include clustering of genes by similarity in expression profiles; searching promoter sequences within clusters for statistically significant TREs; searching TRE databases to identify which gene products bind to specific TREs as TFs; and nonlinear model identification that assembles nuclear connectivity, expression levels of target genes and TFs, and half-life data into a gene regulatory network model. The final result, the gene regulatory network model, is indicated in the bold oval with the heavy line.

8.1 Nuclear connectivity determination

Nuclear connectivity determination requires as inputs gene expression data, promoter sequences, databases of TREs, and protein-DNA interaction data, and involves clustering, searches for significant TREs in the clusters, and databases/literature searches for the TF/TRE couplings. It is the most complex aspect of the gene network identification problem described presently. The primary assumption is that genes with similar expression profiles are regulated similarly, and thus the promoters of genes clustered by gene expression profiles will be enriched for the TREs to which the involved TFs bind and regulate gene expression. Details of each of the steps are provided below.

8.1.1 Clustering

In the present work, the objective of clustering genes is to partition them into groups that are regulated similarly, so that the promoters of the genes in the same group may be searched for over-represented TREs. The assumption that genes that are regulated similarly will have similar expression profiles allows genes to be grouped on the basis of similarities in their expression time courses (synexpression groups, Niehrs and Pollet, 1999). Many approaches for clustering gene expression data have been described (Sherlock, 2000; Dougherty et al., 2002).

8.1.2 TRE search

Given the gene clusters, the next step in determining nuclear connectivity is to search the promoters of the genes in the group for TREs that are statistically over-represented ("enriched") compared to random groups of the same size (Bucher, 1999; Altman and Rayachaudhuri, 2001). Given that TREs are short and degenerate, they are likely to appear randomly in practically any DNA sequence, and thus the mere presence of a TRE in a promoter is not strongly indicative of the regulation of that gene by the TF that binds to the TRE. The presence of a statistically over-represented TRE in a group of promoters obtained by clustering expression profiles, however, does strongly suggest that the binding of a TF to that TRE is responsible in some way for the particular pattern of expression in that cluster. It follows that clustering plus TRE search may be viewed as a means to identify both the TREs that are actively bound during a particular process, as well as biologically significant TRE-gene pairings.

The first step of the TRE search is to obtain the promoter sequences for the genes in the various clusters. Several bioinformatics tools and databases may do this automatically given a gene list, including SCPD (Zhu and Zhang, 1999) for yeast, and PAINT (Vadigepalli et al., 2003) for mammals. Given the gene clusters and their respective sets of promoter sequences, there are two general methods for finding TREs. One method attempts to discover over-represented TREs *à priori* from the promoter sequences themselves, without any prior knowledge, and has been implemented in several bioinformatics tools (Roth et al., 1998; Tavazoie et al., 1999). Another method involves a two-step process in which the promoter sequences are searched for TREs that are compiled in databases, such as SCPD (Zhu and Zhang, 1999), or TRANSFAC (Matys et al., 2003). Tests of statistical significance for the enrichment of TREs in each of the clusters are then performed (Elkon et al., 2003; Vadigepalli et al., 2003; Zak et al., manuscript in preparation). In some cases, it may also be possible to test the TREs identified *à priori* for enrichment in each of the clusters.

Were TREs only to appear once per promoter sequence, the test for their statistical enrichment could be readily carried out using the hypergeometric distribution (Jakt et al., 2001). TREs however, are known to occur in multiple copies in individual promoters (Taylor et al., 2000), and repeat appearances must be factored into the test for significance because repeated TREs are less

likely to have arisen randomly. One group has attempted to ameliorate this problem by using an extended hypergeometric distribution in the significance test (Elkon et al., 2003), but this approach becomes intractable for numbers of TRE/promoter sequence greater than about 3 (we observed as many as 40 TREs/500 base pair promoter sequence). An alternative is to use an empirical approach in which reference distributions for the significance test are constructed by randomly sampling clusters from total gene population and tabulating the number of times each TRE was observed in each cluster (Zak et al., manuscript in preparation). The reference distributions are then used to directly calculate the probability (p-value) that the observed number of occurrences of each TRE in each cluster can be explained by random variation alone. TREs with very low p-values are taken to be statistically significant, with typical cutoffs being $p < 0.05$ (5%) or $p < 0.01$ (1%).

8.1.3 Assembling nuclear connectivity

The combined result of the clustering and TRE search steps is both a list of TREs that are likely to be actively bound in the system *and* a network of TRE-gene pairings. This TRE-gene connectivity is not sufficient for modeling, however, because the TREs must be related back to the TFs that bind to the TREs. In some cases this is straightforward, with the TRE being specific for a single TF. It is more common, however, for TF heterodimers to bind to TREs, with the TF dimerization partners determining how they regulate the transcription of target genes (Alberts et al., 1994). Information concerning the TFs that bind to specific TREs may be obtained from databases (Matys et al., 2003) or the literature. When the TFs specific for the TREs have been identified the nuclear connectivity determination is essentially complete. Protein-DNA interaction data (Lee et al., 2002), if available, may be used to filter out TF-gene interactions that have not been observed experimentally, thereby providing additional refinement to the nuclear connectivity. Although it is appealing to base nuclear connectivity entirely on protein-DNA interaction data, this may not be advantageous given that potential binding of a TF to a promoter, as provided in protein-DNA interaction data, is not indicative of the activity of that TF during the process of interest. Combining protein-DNA interaction data with predicted TF activities from gene expression clustering/TRE is preferable because only the TFs that are likely to be active are considered in the subsequent steps of the identification.

8.2 Model Identification

Once the nuclear connectivity is known, the dynamic gene regulatory network model may be identified from the gene expression data for the regulated genes and the TFs. This involves specification of the dynamical model structures and parameter estimation for the nuclear ($\mathbf{g}(\cdot)$) and cytoplasmic ($\mathbf{h}(\cdot)$) models (Figure 2(b)). The gene regulatory network model identification technique used in the present work has been described elsewhere (Zak et al., 2003b). It is based on the Hartley Modulating Functions (HMF) approach to continuous-time system identification (Patra and Unbehauen, 1995; Daniel-Berhe and Unbehauen, 1999).

Modulating functions (MF) approaches were developed by Shinbrot (1957) as a means to estimate parameters in nonlinear dynamical systems by linear regression. The key steps in MF approaches are (1) expressing the system in input-output differential (IOD) form that is linear in the parameters (LP) to be estimated, (2) multiplying both sides of the IOD system by the known, smooth MF $\phi(t)$, (3) integrating the system from $t=0$ to $t=T$ (the final sampling time), and (4) applying integration by parts to transfer derivatives of the states to derivatives of $\phi(t)$. Appropriate selection of $\phi(t)$ removes any need to estimate derivatives from data, while performing steps 1-4 with several MFs allows the model parameters to be estimated by linear

regression. MF approaches have the advantages of simplicity in parameter estimation, no need to approximate derivatives from data, and, given that the experimental data is integrated, they relax the traditional requirement for uniformly sampled data. Since uniformly sampled data is uncommon in biology, this latter point is of particular interest for the current discussion. The primary disadvantages are that the system must be LP when expressed in IOD form, and that bias-free parameter estimates cannot be guaranteed for all nonlinear models (Niethammer et al., 2001).

Possible dynamical structures for the nuclear (**g**(·)) and cytoplasmic (**h**(·)) models are now described. For **g**(·), the dependence of the expression level of gene *i* on the activity of a single TF, *u*(*t*), can be described by a simple linear model:

$$\dot{x}_i = g(u(t), x_i(t), p_{g_i}) = a_i u(t) - d_i x_i(t) \tag{3}$$

where $x_i(t)$ is the scaled ($-1 \le x_i(t) \le 1$, $x_i(0) = 0$) mRNA level for gene *i*, a_i is the transcriptional activity constant for gene *i* ($a_i > 0$ corresponds to activation, $a_i < 0$ corresponds to repression), and d_i is the first order degradation constant for x_i. There are several ways to model genes that are regulated by more than one TF, as the TFs may have additive, multiplicative, or complex effects on the transcription rate. Choosing between the options will have to be based on the biology of the system, or else multiple alternatives should be considered.

Several dynamical structures may be considered for the cytoplasmic model **h**(·). We presently restrict our attention to autonomous systems (for example, the cell cycle) for which the external inputs, **v**(*t*), are zero, and TF activity is regulated at the transcriptional level, and thus *u*(*t*) depends only on the TF mRNA levels. In the simplest case, the TF is composed of only one protein, and thus depends only on one mRNA level, giving:

$$u(t) = h(x(t), p_g) = q(x_{TF}(t)) \tag{4}$$

where $x_{TF}(t)$ is the scaled ($0 \le x_{TF}(t) \le 1$, $x_{TF}(0) \neq 0$) TF mRNA concentration, and $q(\cdot)$ is a nonlinear saturating function, with $0 \le q(\cdot) \le 1$. In the case where the active TF is composed of two proteins, a possible model structure is:

$$u(t) = q(x_{TF_1}(t) \times x_{TF_2}(t)) \tag{5}$$

where x_{TF1} x x_{TF2} is the scaled product of the mRNA concentrations of the two genes that make up the TF. Using Equation 5, the expression profiles of two genes are combined to create a single time course of TF activity.

The models in Equations 4 and 5 may be made more realistic by including the delay that occurs between the appearance of the mRNA of a gene and functional protein, due to several intermediate biological processes (translation, for example). These additional biochemical processes may be approximated by increasing the dynamical order of Equations 4 and 5, respectively as follows:

$$\dot{u} = q(x_{TF}(t)) - eu(t) \tag{6}$$

$$\dot{u} = q(x_{TF_1}(t) \times x_{TF_2}(t)) - eu(t) \tag{7}$$

where *e* is the first order degradation constant for the active TF. The models in Equations 6 and 7 will be referred to as *lagged* models (because they incorporate an additional dynamic lag),

while those in Equations 4 and 5 will be referred to as *unlagged* models. By including the lag, **h**(·) becomes a dynamic model, in contrast to the static models of Equations 4 and 5.

In summary, for each gene *i*, two parameters (*a* and *d*) are to be estimated in **g**(·), and either zero or one parameter (lagged models) needs to be estimated in **h**(·), given prior knowledge of *q*(·). The function *q*(·) is often not known, and is thus an additional model structure to define with additional parameters to estimate. One possible strategy is to define a canonical structure and a nominal set of parameter values that correspond to different qualitative behaviors. The model parameters that do not appear in *q*(·) are then estimated for each nominal value in the set. In the present work, *q*(·) is a single parameter nonlinear saturating structure, and three parameter values, corresponding to approximate linearity, weak nonlinearity, and strong nonlinearity, are considered (details of the approach given in Zak et al., 2003b).

Equation 3, by explicitly including mRNA degradation rate constants, also allows this other set of data to be included into the overall structured gene regulatory network identification approach. Genome-wide measurements of mRNA half lives (and therefore degradation constants) are increasingly available (Wang et al., 2002; Fan et al., 2002; Selinger et al., 2003; Yang et al., 2003), and may be readily integrated into the model identification through parameter d_i. Including this information reduces the number of parameters to be estimated, and, given that the time scale on which any mRNA level responds to changes in transcription is determined by its half-life (Hargrove and Schmidt, 1989), is a means to include *dynamic* constraints in the modeling approach.

We must point out that the processes encapsulated in function **h**(·) are generally highly dynamic and highly regulated in cells, and it is possible that the models in Equations 4-7 are overly simple approximations. Genomic and functional genomic data types are largely specific to transcriptional regulation, however, and thus provide little information for characterizing **h**(·). For this reason, **h**(·) depends the most strongly on what system-specific prior knowledge is available. When little is known about regulation of TF activity, it is reasonable to assume that it occurs at the transcriptional level, and thus Equations 4-7 are appropriate. In cases where information about more complex modes of regulation is available, **h**(·) can be made more complex as necessary. For example, it is straightforward to integrate computational models that describe how regulators of TF activity are regulated by extracellular signals and/or intracellular signaling (Ramkrishnan et al., 2002; Neves and Iyengar, 2002; Bhalla, 2003), or how TF activity is regulated in specific cellular process (Chen et al., 2002), into **h**(·). One demonstration of this integration may be found in Jin et al. (2003), although the authors do not use a structured modeling approach in their study.

8.3 Case Study: Yeast Cell Cycle

The yeast cell cycle is an attractive system for demonstrating the approach of the present work for many reasons. These include the availability of several microarray time courses with enough time points (~15) to render them suitable for modeling (Cho et al., 1998; Spellman et al., 1998); the availability of half-lives for nearly every gene in the yeast genome (Holstege et al., 1998; Wang et al., 2002); the availability of protein-DNA interaction data for most of the yeast TFs (Lee et al., 2002); and the availability of a database (SCPD) from which promoter regions and putative TREs for nearly every yeast gene may be obtained (Zhu and Zhang, 1999). In the present study, we restricted our analysis to genes expression profiles of genes from Cho et al.

(1998) that were most variable as defined by Tavazoie et al. (1999) and had transcript half life data available in Wang et al. (2002), 2042 genes in total.

8.3.1 Nuclear connectivity determination: clustering

We used the k-means clustering algorithm (Hartigan and Wong, 1979) in the present work to cluster the gene expression profiles. The clustering results obtained using this method are sensitive to the clustering parameters (Sherlock, 2000), which include the number of clusters and the initial specification of the cluster centers, but we hypothesize that biologically meaningful results should not be. For this reason, we took an approach that allowed us to identify results that were robust to variations in the clustering parameters. Specifically, we performed the clustering with three different numbers of clusters (10, 30, and 60, following Tavazoie et al., 1999) and five different initializations, giving a total of 15 different clustering results that together comprised 500 overlapping clusters. This set of clusters was then used in the subsequent TRE search, as described below.

8.3.2 Nuclear connectivity determination: TRE search

Because it makes use of the accumulated knowledge from the literature, we used the database-driven approach for the TRE search. The first step was to obtain the promoter sequences for the genes, which we did using the *S. cerevisae* promoter database (SCPD: Zhu and Zhang, 1999). Specifically, we obtained sequences 500 base-pairs upstream from the start codons. We then used basic pattern matching tools built into SCPD to count the number of times the consensus sequences (and their reverse complements) of the ~50 TREs in SCPD appeared in each promoter sequence. For example, "GGATG", the reverse complement of the consensus sequence for GCR1, was found twice in the promoter sequence for YBL072C. Further details about the methods used to identify TREs in the promoter sequences are available upon request. We then used the empirical approach (Zak et al., manuscript in preparation) to test the 500 clusters for enrichment of specific TREs over random groups of genes of the same size. TREs that had a probability of random occurrence of less than 0.1% ($p < 0.001$) were deemed statistically significant. Finally, to guard against TREs that were highly sensitive to the clustering parameters, we retained only those that were significantly enriched in at least 4 out of the 500 total clusters (corresponding approximately to TREs that were significantly enriched in clustering results from 4/5 different initializations).

Some of the TREs were very robust to variations in the clustering parameters. For example, there was always at least one cluster enriched for both SCB and MCB, regardless of the number of clusters or initialization. Similarly, 4/5, 5/5, and 3/5 of the clustering results that used 10, 30, or 60 clusters, respectively, had at least one cluster enriched for both GCR1 and RAP1. Other TREs showed some sensitivity to the number of clusters, for example, 4/5 of the initializations using 10 clusters had at least one cluster significant for SFF, while 4/5 of the initializations using 60 clusters had at least one cluster significant for both ACE2 and SWI5, but SFF was not significant in any of the clustering results using 60 clusters, and ACE2 and SWI5 were not significant together in any clustering results using 10 clusters. To explore further the robustness of the clustering results to the clustering parameters, we investigated the extent of overlap between genes in clusters obtained using different clustering parameters that were enriched for the same TRE. We found that a significant number of genes (54) always appeared in clusters enriched for both SCB and MCB, regardless of clustering parameters. In contrast, there were no genes that consistently appeared in all clusters enriched for the other TREs.

Overall, eight TREs or TRE pairs were significant in at least four clusters, and 100 out of the 500 total clusters were significantly enriched for at least one TRE. The eight significant TREs/TRE pairs and the number of clusters that were enriched for them are given in Table 3. With two exceptions (out of 6), all TREs that were present in SCPD and were identified in Tavazoie et al. (1999) for the same dataset were also found to be significant in the present study. Additionally, we identified physiologically relevant co-occurrence of TREs that Tavazoie et al. (1999) did not, such as RAP1-GCR1 and ACE2-SWI5 (where the '-' is used to distinguish clusters that were enriched for both TREs from clusters that were enriched for either TRE individually). RAP1 and GCR1, for example, are known to form a complex to regulate ribosomal gene expression (Deminoff and Santangelo, 2001). Additionally, SFF, a key cell cycle TRE, was significant in many clusters of the present results, while it was not in the study by Tavazoie et al. (1999).

As a final step in the TRE search, we compared the centers of clusters that were enriched for the same TRE or pair of TREs. Our objective in doing so was to validate the assumption that genes with similar expression profiles are regulated similarly and thus have common TREs in their promoters. Representative results are shown in Figure 27. Overall, we observed that the centers of clusters enriched for the same TRE were highly similar, which strongly suggests that activity at the enriched TRE is responsible for the variations in gene expression of the member genes. We also observed that centers of clusters enriched for different TREs were noticeably different, demonstrating how activity at different TREs leads to differential regulation of gene expression. Taken together these results validate our assumptions and approach.

Table 3: Number of clusters in which specific TREs were statistically over-represented.

TRE	Number of clusters[a]	Number of genes[b]	Number of genes verified[c]	% Verified
RAP1[e]	10	191	31	16.2%
RAP1-GCR1	9	342	51	14.9%
ACE2-SWI5	4	66	12	18.2%
SFF	12	303	26	8.6%
STRE[e]	15	270	1	0.4%
STRE-MCB^{-1} [d]	11	579	3	0.5%
SCB-MCB[e]	27	54	17	31.5%
MCM1	13	6	0	0.0%

(*a*) Out of a total of 500 clusters.
(*b*) For MCM1 and SCB-MCB clusters, the number of genes is the number of genes that are shared by all clusters with that TRE (intersection). For the other TREs, the intersection between all clusters was zero, therefore the number of genes shown is for the union of all genes found in clusters that were statistically over-represented for that particular TRE.
(*c*) *Verified* indicates that protein-DNA interaction data (Lee et al., 2002) showed that the promoter of the gene was bound by at least one of the TFs that binds to the TRE.
(*d*) "STRE-MCB^{-1}" indicates clusters that were enriched for STRE TREs and depleted of MCB TREs.
(*e*) TREs or TRE pairs that were found in Tavazoie et al. (1999).

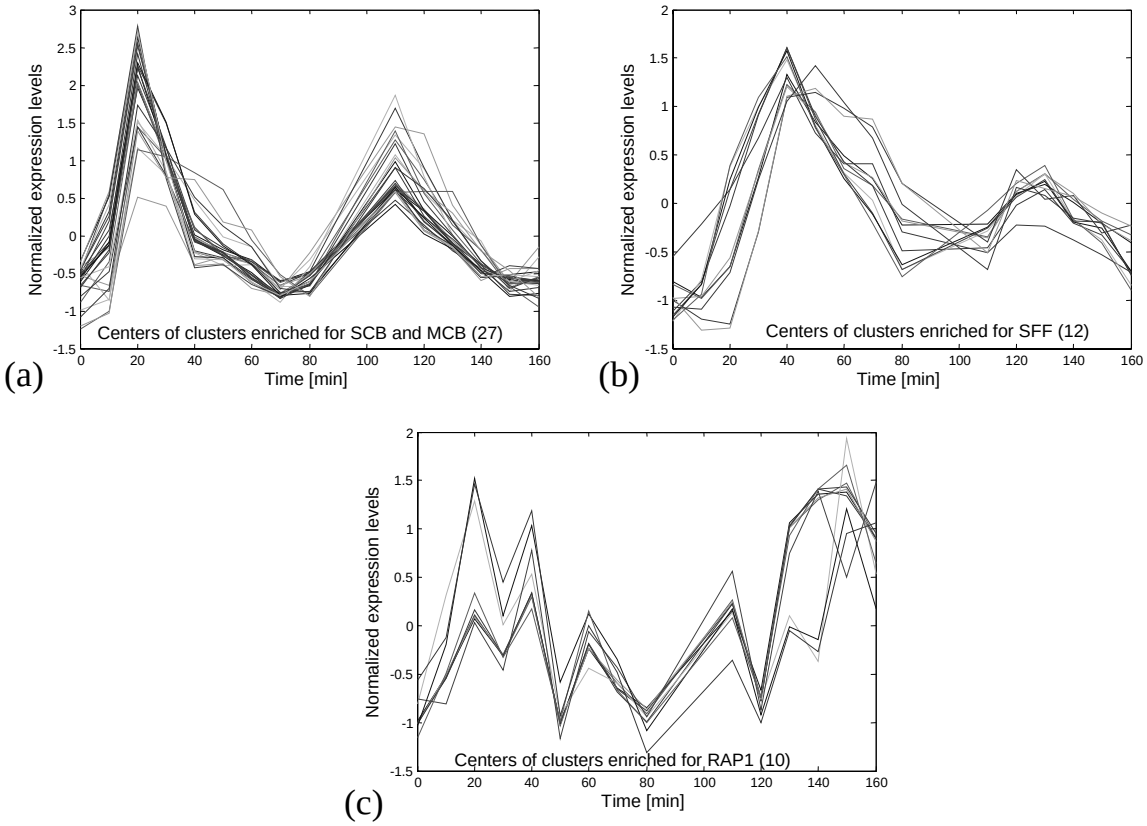

Figure 27: Centers of clusters enriched for the same TRE (or pair of TREs). (a) Centers of clusters (27) in which the TREs SCB and MCB were statistically over-represented in the gene promoters. (b) Centers of clusters (12) in which SFF was statistically over-represented. (c) Centers of clusters (10) in which RAP1 was statistically over-represented. From these plots it is clear that centers of clusters enriched for the same TRE are highly similar, while the centers of clusters enriched for different TREs are substantially different. Expression data from Cho et al., 1998; promoter and TRE information from SCPD (Zhu and Zhang, 1999).

8.3.3 Assembling nuclear connectivity

The final step in the determination of nuclear connectivity was to link the TREs back to the TFs that bind them. We accomplished this by means of a literature search. SCB and MCB are bound by the transcription factor SBF, a complex of SWI4 and SWI6, and MBF, a complex of MBP1 and SWI6, respectively (Taylor et al., 2000). Since the clusters that were enriched for SCB were also enriched for MCB, the genes contained in these clusters were treated as targets of SWI4 or MBP1 individually, or as targets of the product of SWI4 and MBP1 in the subsequent model identification. SFF is largely bound by a complex of MCM1 and FKH2 (Kumar et al., 2000) and thus genes in clusters enriched for SFF treated as targets of MCM1 or FKH2 individually or the product of MCM1 and FKH2. STRE is bound by MSN2 and/or MSN4 (Schmitt and McEntee, 1996) and for this reason STRE cluster genes were treated as targets of MSN2, MSN4, or the product of MSN2 and MSN4. Although RAP1 is known to bind to promoters in a complex with GCR1 (Deminoff and Santangelo, 2001), for simplicity, genes that were in clusters enriched for RAP1 were modeled as targets of RAP1 individually. Finally, ACE2 and SWI5 are known to regulate genes individually or jointly (Doolin et al., 2001) and thus the ACE2-SWI5 genes were treated targets of ACE2, SWI5, or the product of ACE2 and SWI5 in the subsequent modeling.

Before proceeding to the model identification, we refined the nuclear connectivity by retaining only those TF-gene links that have been observed in the protein-DNA interaction data of Lee et al. (2002). The results are given in Table 3, where verified genes were those for which at least

one predicted TF-gene binding interaction was observed by Lee et al. (2002). It is apparent that the results were largely TF dependent, ranging from 30% verification of interactions for the SCB-MCB genes, ~20 per cent for RAP1 and ACE2-SWI5 genes, ~10 per cent for SFF genes, and ~0 per cent for STRE genes. For the subsequent modeling, we used only the genes for which TF binding was verified. Since none of the STRE genes were verified, an exception was made for STRE by randomly selecting 30 genes for model identification.

8.3.4 Model identification

Given the nuclear connectivity, it was possible to proceed to the model identification, which involved parameter estimation for, and selection between, the various dynamical model structures in Equations 2-6. We performed model identification for only one loop of Figure 2(b). In other words, we only modeled how variation in the expression of the TFs leads to variation in the expression of their target genes. We did not model how the TFs regulate the expression of the other TFs, although this is possible. We used the HMF method to estimate the parameters, which is appropriate for the gene expression data used in the present study because it was rendered asynchronous by poor hybridization at time points 90 and 100 minutes that led these data to be left out (Tavazoie et al., 1999). For the mRNA degradation rate constant, d_i, we considered the minimum, mean, and maximum values for each gene given by Wang et al. (2002). To add flexibility to the model identification, both the lagged and unlagged versions of the cytoplasmic model were used for each gene, and three parameterizations of $q(\cdot)$, corresponding to linear, weakly nonlinear, and strongly nonlinear, were used. In summary, a total of 18 models were identified for each TF/target gene pairing. To investigate the importance of the prior knowledge of nuclear connectivity, we additionally modeled the ACE2-SWI5 target genes as targets of 10 other genes randomly selected from the set of all genes. To quantify the success of the model identification, we defined as *well-modeled* gene-TF pairings that had a sum of squared errors (SSE) between the experimental data and model prediction that was small on an absolute scale (SSE < 1.5), and on a relative scale in comparison to the SSE that would be obtained from a purely correlative model between the TF and the target gene (SSE(model)/SSE(correlation) < 0.66). By these criteria, *well-modeled* described identification results that were highly suggestive of causal links between the TFs and their target genes.

8.3.5 Model identification results

The results of the model identification are summarized in Table 4, where the number of genes that met the well-modeled criteria for each TF is indicated, along with whether the TF was found to be an activator or repressor of its targets, and any other systematic trends in the dynamical model structures (extent of nonlinearity, lag or no lag) of the TF target pairings that “modeled well”. A complete listing of the identified model parameters is given in the online appendix (http://www.dbi.tju.edu/dbi/publications/cache04). With the exception of the pairing of the ACE2-SWI5 genes as targets of random genes, all the genes that were *well-modeled* as targets of the same TF were either uniformly activated or repressed by that TF, suggesting that the TFs play specific roles as activators or repressors in the present system.

Similarly to the verification of TF/target gene pairings with the protein-DNA interaction data, the number of genes that were *well-modeled* depended strongly on the gene group and the particular TF. The SCB-MCB genes had the largest fraction that were well-modeled, with 70% of the genes verified as being bound by either SWI4 or MBP1 being well-modeled as activated targets of the product of SWI4 and MBP1, and 60 per cent of all the genes that were common to all SCB-MCB clusters being *well-modeled* as activated targets of SWI4. Examples of these genes are shown in Figure 28. Additionally, some genes were *well-modeled* as repressed targets of

MBP1, with a small subset of those also being well-modeled as activated targets of SWI4 and/or the product of SWI4 and MBP1 (example gene shown in Figure 28). This result suggests that SWI4 and MBP1 may play opposing roles in regulating the expression of the SCB-MCB genes. Finally, the SCB-MCB genes were best modeled using the lag-free model, with the linear model being optimal for the genes modeled as targets of SWI4 and the product of SWI4 and MBP1, and the strongly nonlinear model being best for the targets of MBP1.

The SFF genes had the second highest fraction that fit the criteria for being *well-modeled,* with 40 per cent as activated targets of MCM1, and 30 per cent as repressed, lagged, targets of FKH2. Importantly, the sign of the TF first order degradation constant (*e*) in the lagged models was negative for the genes *well-modeled* as targets of FKH2. This result weakens the case for a causal transcriptional link of the type postulated in Equation 5 between FKH2 and its targets, and suggests that a more complex model for FKH2 activity may be necessary. Interestingly, unlike the SCB-MCB genes, the smallest fraction of SFF genes were well-modeled as targets of the product of MCM1 and FKH2. Similar to the SCB-MCB genes, there were SFF genes that were *well-modeled* either as activated targets of MCM1 or repressed targets of FKH2. Smaller fractions of the RAP1, STRE, and ACE2-SWI5 genes were well-modeled as compared to the SCB-MCB and SFF genes, suggesting that the activities of these TFs may be regulated post-transcriptionally, requiring more complex structures for the cytoplasmic model $\mathbf{h}(\cdot)$. This is especially true for the STRE genes, for even those that were modeled-well as targets of MSN2 and MSN4 gave rise to negative estimates for the parameter *e*, that, similarly to the SFF genes well modeled as targets of FKH2, suggests a more complex mode of transcriptional regulation.

The results of modeling the ACE2-SWI5 genes as targets of randomly selected genes clearly demonstrated the importance of prior knowledge of nuclear connectivity in gene regulatory network modeling. Comparable, and sometimes much greater, percentages of the ACE2-SWI5 genes were *well-modeled* as targets of the random genes than their true regulators, ACE2 and SWI5. Some of the genes, such as PCK1, KTR2, and LYS4, are involved in cellular processes that are very different from transcriptional regulation (TCA cycle, protein glycosylation, and lysine biosynthesis, respectively), and thus it is highly unlikely that the good identification results are anything other than artifact. Illustrative examples are shown in Figure 29, where genes that are known to be targets of ACE2 and SWI5, individually or jointly, are *well-modeled* as targets of random genes.

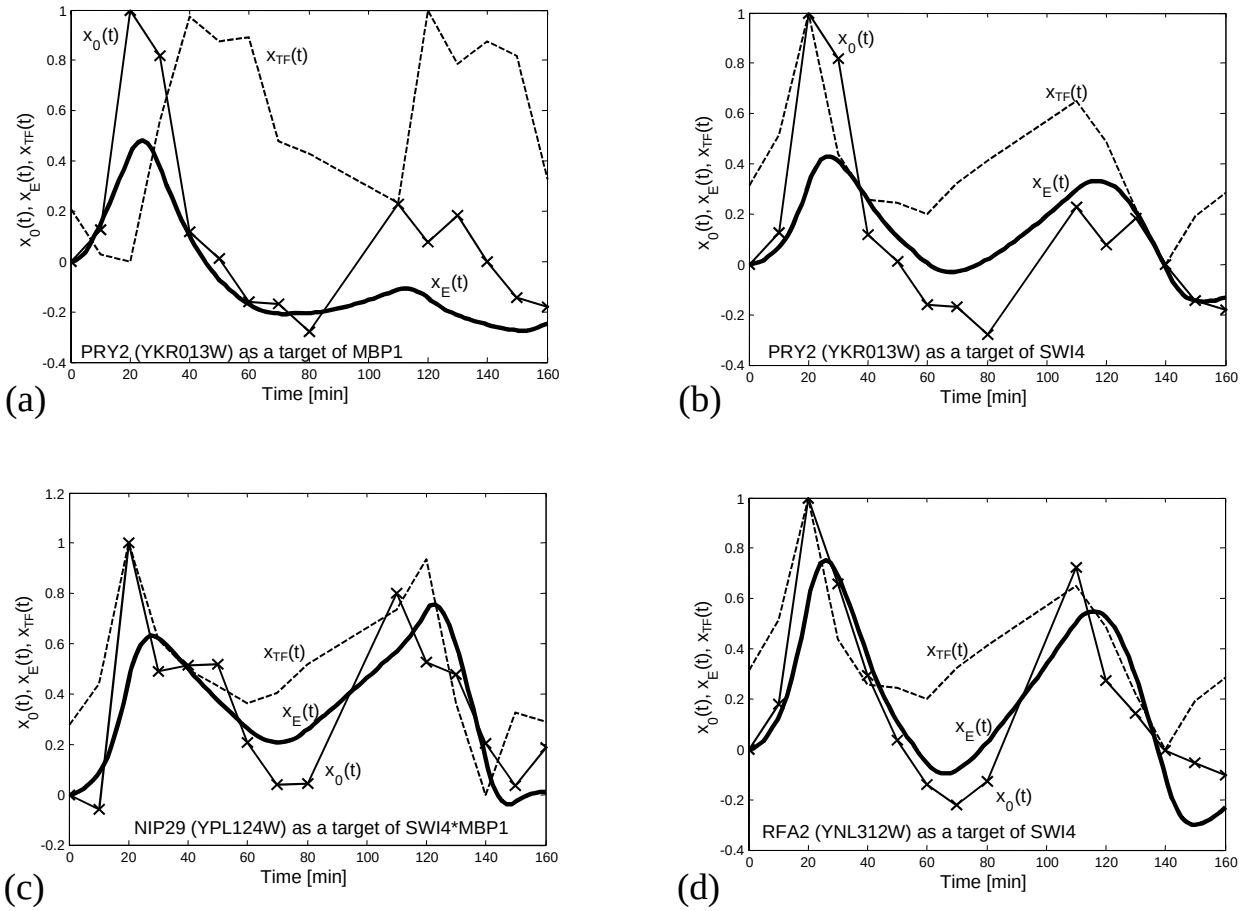

Figure 28: Representative identification results for genes modeled as targets of MBP1 (binds to MCB), SWI4 (binds to SCB), or the product of SWI4 and MBP1. (a and b) Example of a gene that is well-modeled as a repressed target of MBP1 (a) or an activated target of SWI4 (b). (c) Example of a gene best modeled as an activated target of the product of SWI4 and MBP1. (d) Example of a gene best modeled as an activated target of SWI4 alone. Dashed lines: $x_{TF}(t)$, the scaled mRNA level of the TF (or scaled product of the mRNA levels of the TFs) that regulates the gene. Solid line, crosses: $x_0(t)$, the experimental expression level of the regulated gene. Thick, smooth line: $x_E(t)$, the expression level of the gene predicted from the model identification.

Table 4: Model identification results: (Table 4 caption on following page)

RAP1

TF	Number of Genes	%Genes	Nonlinearity	Lag	Activator/Repressor
RAP1	5	16.1%	~	yes	activator

ACE2-SWI5

TF	Number of Genes	%Genes	Nonlinearity	Lag	Activator/Repressor
ACE2	2	16.7%	~	yes	activator
SWI5	3	25.0%	~	~	activator
ACE2*SWI5	3	25.0%	~	~	activator

SFF

TF	Number of Genes	%Genes	Nonlinearity	Lag	Activator/Repressor
FKH2	8	30.8%	strong	yes	repressor
MCM1	11	42.3%	~	~	activator
MCM1*FKH2	5	19.2%	strong	no	activator

STRE

TF	Number of Genes	%Genes	Nonlinearity	Lag	Activator/Repressor
MSN2	5	16.7%	strong	yes	activator
MSN4	3	10.0%	strong	yes	activator
MSN2*MSN4	0	0.0%	~	~	~

SCB-MCB (verified)

TF	Number of Genes	%Genes	Nonlinearity	Lag	Activator/Repressor
MBP1	4	23.5%	strong	~	repressor
SWI4	10	58.8%	linear	no	activator
MBP1*SWI4	12	70.6%	linear	no	activator

SCB-MCB (all)

TF	Number of Genes	%Genes	Nonlinearity	Lag	Activator/Repressor
MBP1	7	13.0%	strong	no	repressor
SWI4	33	61.1%	linear	no	activator
MBP1*SWI4	22	40.7%	linear	no	activator

ACE2-SWI5 (rand)

TF	Number of Genes	%Genes	Nonlinearity	Lag	Activator/Repressor
YDR012W	1	8.3%	strong	no	repressor
YLR395C	1	8.3%	weak	yes	repressor
PCK1	5	41.7%	~	yes	repressor
YNL114C	8	66.7%	strong	~	~
YPR182W	1	8.3%	weak	yes	repressor
KTR3	3	25.0%	~	~	activator
YIR043C	6	50.0%	~	yes	activator
YNR073C	8	66.7%	strong	no	repressor
YKL161C	2	16.7%	linear	no	activator
LYS4	2	16.7%	weak	yes	activator

Table 4: Model identification results: (see above) *Number of genes* is the number of genes that were well modeled as targets of the TFs that bind to TREs that were statistically over-represented in the clusters of which the genes were members. *Nonlinearity* indicates the degree of nonlinearity that gave the smallest SSE for the genes that were well modeled as targets of the TF. *Lag* indicates whether or not the lagged model gave the smallest SSE for genes that were well modeled as targets of the TF. *Activator/Repressor* indicates whether the TF was found to be a transcriptional activator or transcriptional repressor of the genes modeled as targets of the TF. "~" indicates that the results were gene dependent.

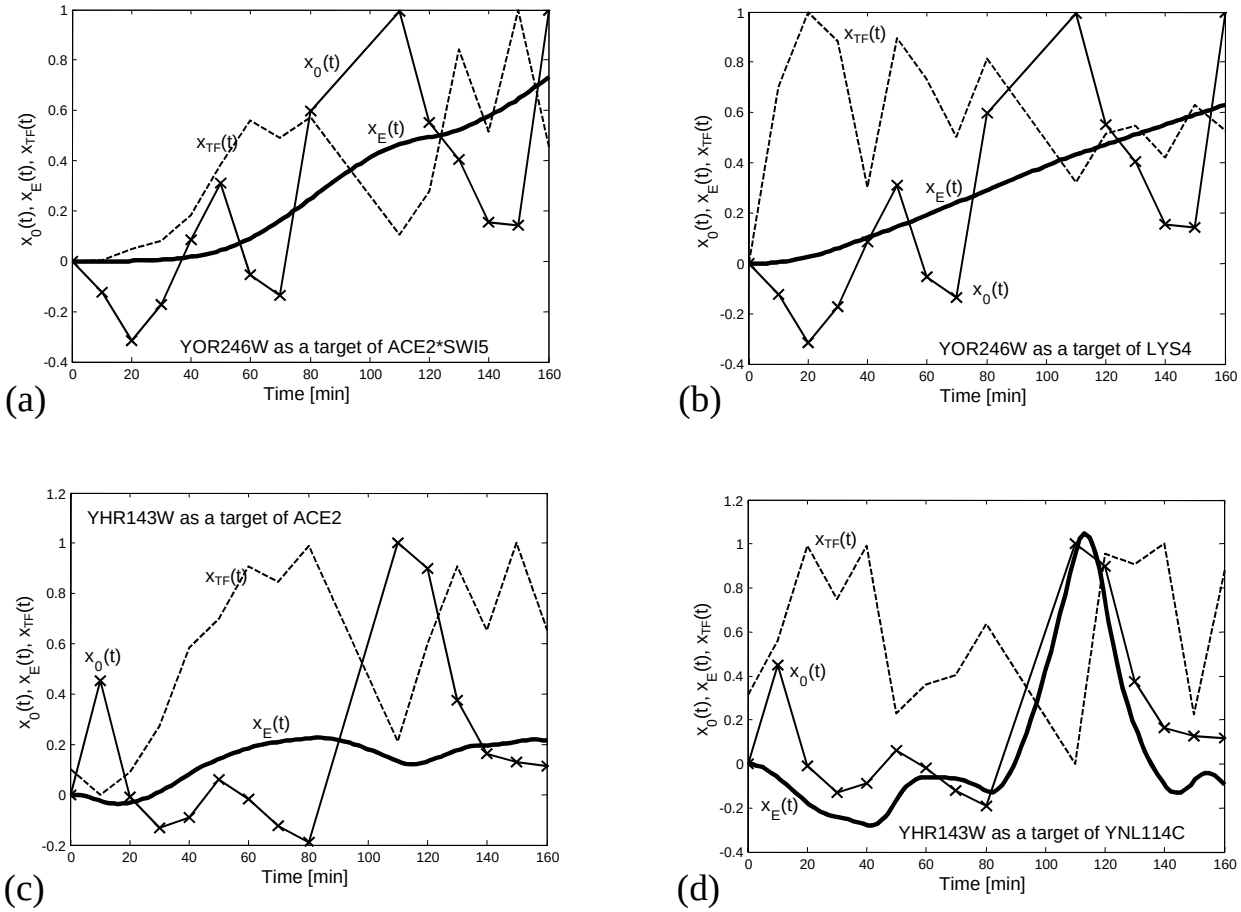

Figure 29: Representative results of modeling targets of ACE2/SWI5 as targets of ACE2/SWI5 or random genes. (a and b) Example of a gene that is known to be jointly regulated by ACE2 and SWI5 (Doolin et al., 2001) that is reasonably well modeled as a target of the product of ACE2 and SWI5 (a) or a random gene (b), in this case, a member of the lysine biosynthesis pathway. (c and d) Example of a gene known to be regulated by ACE2 (Doolin et al., 2001) that is not well modeled as a target of ACE2 (c) but is very well modeled as a target of a random gene (d), in this case a gene with unknown function. Dashed lines: $x_{TF}(t)$, the scaled mRNA level of the TF (or scaled product of the mRNA levels of the TFs) that regulates the gene. Solid line, crosses: $x_0(t)$, the experimental expression level of the regulated gene. Thick, smooth line: $x_E(t)$, the expression level of the gene predicted from the model identification.

9 Conclusions

In the present project, we have demonstrated how systems engineering approaches that explicitly recognize the complexities, constraints, and idiosyncrasies of biological systems can effectively handle complex biological problems. We described how imposing structure via fundamental knowledge and the inclusion of multiple types of data into the resulting structured modeling and identification approach can render an otherwise intractable problem more tractable (although this is achieved at the expense of introducing additional idiosyncrasies specific to each type of additional data). Given the current rate of progress in genomic sequencing, annotation, and bioinformatics tool development, additional data types and information that may constrain model structures are increasingly available. There is no reason to exclude these data types from attempts to model gene regulation because they are available for practically any system of interest. While the data collections for all organisms are not currently as extensive as they are for yeast, the information that is available can nevertheless significantly enhance the modeling efforts. For example, merely specifying that only TFs may regulate the expression of genes can reduce the number of model parameters by an order of magnitude. Using data that is currently available for yeast, we demonstrated a framework for integrating multiple data types into subcellular and nuclear connectivity structures that may be used as prior knowledge in the modeling and identification of gene regulatory networks. Nuclear connectivity, as obtained through a structured modeling approach, specifies which genes are regulated by which TFs and can greatly improve the tractability of the gene network identification problem.

In case studies 1 through 4, PAINT, in combination with experimentally associated genes list and genomic sequence data, has identified the TREs and cognate TFs likely to subserve the biological regulation studied in each case. These results are discovered in a scalable and automated manner using a bioinformatics approach to analyze the data from global methods such as microarrays, ChIP, etc. The primary purpose of PAINT is to provide a scalable and extensible platform to automate the process of mining the existing databases for known regulatory information for a large number of genes of interest in a particular experiment or analysis. The interaction matrix generated represents candidate connections in the regulatory network. In a particular experiment, only a subset of transcription factors in the cell is active. The over-represented TREs identified from CIM indicate a set of TREs that are likely to be active. Time series data of TF activity from ChIP or promoter binding assays provides a set of active TFs. By combining these two sets together, the most likely regulators in that particular experimental context are obtained. Combining this data with the interaction matrix from PAINT, a smaller subset of interaction matrix that represents the candidate network specific to that particular experimental perturbation can be constructed.

The experimental and computational methods presented here identify a set of genes and transcription factors that are significant in understanding the function of the gene regulatory network in question. As demonstrated in the Case study 5, this network structure information can be directly utilized in construction of an *in silico* model of the regulatory network. Incorporation of this model into simulations along with models of signaling pathways and electrophysiology is the key to analyzing the immediate, intermediate and long-lasting cellular response to an external signal.

10 References

Aeberhard S., D. Coomans, and O. de Vel, Comparative-analysis of statistical pattern-recognition methods in high-dimensional settings. *Pattern Recognition*, 27(8):1065-1077, 1994.

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K. & Watson, J.D., *Molecular Biology of the Cell*. New York: Garland Publishing, Inc., 1994.

Altman, R.B., & Raychaudhuri, S., Whole-genome expression analysis: challenges beyond clustering. *Curr. Opin. Struct. Biol.*, **11**(3), 340-7, 2001.

Arnone, M.I. & Davidson, E.H., The hardwiring of development: organization & function of genomic regulatory systems. *Development*, **124**(10), 1851-64, 1997.

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., & Sherlock, G., Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**(1), 25-9, 2000.

Baraldi A. and L. Schenato, Soft-to-hard model transition in clustering: a review, Technical Report 99-010, Berkeley, California, 1999.Baxevanis, A.D. The Molecular Biology Database Collection: 2003 update. *Nucleic Acids Res.*, **31**(1), 1-12, 2003.

Bhalla, U.S., Understanding complex signaling networks through models & metaphors. *Prog. Biophys. Mol. Biol.*, **81**(1), 45-65, 2003.

Bloch K.M. and G. R. Arce, Nonlinear correlation for the Analysis of Gene Expression Data. In *10th International Conference on Intelligent System for Molecular Biology* (ISMB), Canada, 2002.

Brazhnik, P., de la Fuente, A., & Mendes, P., Gene networks: how to put the function in genomics. *Trends Biotechnology*, **20**(11), 467-72, 2002.

Brivanlou, A.H., & Darnell, J.E. Jr., Signal transduction & the control of gene expression. *Science*, **295**(5556), 813-8, 2002.

Bucher, P., Regulatory elements & expression profiles. *Curr. Opin. Struct. Biol.*, **9**(3), 400-7, 1999.

Bussemaker, H.J., Li, H., & Siggia, E.D., Regulatory element detection using correlation with expression. *Nat. Genet.*, **27**(2), 167-71, 2001.

Camon, E., Magrane, M., Barrell, D., Binns, D., Fleischmann, W., Kersey, P., Mulder, N., Oinn, T., Maslen, J., Cox, A., & Apweiler, R., The Gene Ontology Annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, & InterPro. *Genome Res.*, **13**(4), 662-72, 2003.

Chakaravathy S.V. and J. Ghosh, Scale based clustering using a radial basis function network. *IEEE Transactions on Neural Networks*, 2(5):1250-1261, 1996.

Chen, K.C., Csikasz-Nagy, A., Gyorffy, B., Val, J., Novak, B., and Tyson, J.J., Kinetic analysis of a molecular model of the budding yeast cell cycle. *Mol. Biol. Cell.*, **11**(1), 369-391, 2000.

Chen, T., He, H.L., & Church, G.M., Modeling gene expression with differential equations. *Proc. Pac. Symp. Biocomp.*, 4, 29-40, 1999.

Cho, R.J., Campbell, M.J., Winzeler, E.A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T.G., Gabrielian, A.E., Landsman, D., Lockhart, D.J., & Davis, R.W., A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, **2**(1), 65-73, 1998.

Csete, M.E., & Doyle, J.C., Reverse engineering of biological complexity. *Science*, **295**(5560), 1664-9, 2002.

D'Haeseleer, P., Wen, X., Fuhrman, S., & Somogyi, R., Linear Modeling of mRNA Expression Levels During CNS Development & Injury. *Proc. Pac. Symp. Biocomput.*, **4**, 41-52, 1999.

Daniel-Berhe, S., & Unbehauen, H., Physical parameters estimation of the nonlinear continuous-time dynamics of a DC motor using Hartley modulating functions method. *J. Franklin I.*, **336**, 481-501, 1999.

Davies D.L. and D.W. Bouldin, A cluster separation measure. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1: 224-227, 1979.

de la Fuente, A., Brazhnik, P., & Mendes, P., Linking the genes: inferring quantitative gene networks from microarray data. *Trends Genet.*, **18**(8), 395-8, 2002.

De Moor, B., De Gersem, P., De Schutter, B., & Favoreel, W., DAISY: A database for identification of systems. *Journal A.*, **38**(3), 4-5, 1997.

Deminoff, S.J., & Santangelo, G.M., Rap1p requires Gcr1p & Gcr2p homodimers to activate ribosomal protein & glycolytic genes, respectively. *Genetics.*, **158**(1), 133-43, 2001.

D'Haeseleer, P., Liang, S., & Somogyi, R., Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, **16**(8), 707-26, 2000.

Dimitriadou E., A. Weingessel and K. Homik, Voting-merging: an ensemble method for clustering. In *Proceedings of International Conference on Artificial Neural Network*, pp. 217-224, Vienna, 2001.

Doolin, M.T., Johnson, A.L., Johnston, L.H., & Butler, G., Overlapping & distinct roles of the duplicated yeast transcription factors Ace2p & Swi5p. *Mol. Microbiol.*, **40**(2), 422-32, 2001.

Dougherty, E.R., Barrera, J., Brun, M., Kim, S., Cesar, R.M., Chen, Y., Bittner, M., & Trent, J.M., Inference from clustering with application to gene-expression microarrays. *J. Comput. Biol.*, **9**(1), 105-26, 2002.

Duda R.O. and P.E. Hart, *Pattern Classification and Scene Analysis*. New York: John Wiley and Sons. 1973.

Dudoit S. and J. Fridlyand, Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090-1099, 2003.

Eisen M.B., P.T. Spellman, P.O. Brown, and D. Botstein, Clustering analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA* 95:14863-14868, 1998.

Elkon, R., Linhart, C., Sharan, R., Shamir, R., & Shiloh, Y., Genome-wide in silico identification of transcriptional regulators controlling the cell cycle in human cells. *Genome Res.*, **13**(5), 773-80, 2003.

Fan, J., Yang, X., Wang, W., Wood, W.H., Becker, K.G. & Gorospe, M., Global analysis of stress-regulated mRNA turnover by using cDNA arrays. *Proc. Natl. Acad. Sci. USA.*, **99**(16), 10611-6, 2002.

Fickett, J.W., & Hatzigeorgiou, A.G., Eukaryotic promoter recognition. *Genome Res.*, **7**(9), 861-78, 1997.

Fischer B. and J.M. Buhmann, Bagging for path-based clustering. *IEEE Transactions on Pattern Analysis and Machine Intellignece*, 25(11):1411-1415, 2003.

Fisher R.A., The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7(part II): 179-188, 1936.

Fraley C. and A.E. Raftery, How many clusters? Which clustering method? Answers via model-based cluster analysis. *Computer Journal*, 41:578-588, 1998.

Fred ALN, and A. K. Jain, Data clustering using evidence accumulation. In *16th International Conference on Pattern Recognition* (ICPR'02), pp. 276-280, Canada, 2002.

Gardner, T.S., di Bernardo, D., Lorenz, D., & Collins, J.J., Inferring genetic networks & identifying compound mode of action via expression profiling. *Science*, **301**(5629), 102-5, 2003.

Gluck M.A. and J.E. Corter, Information, uncertainty and the utility of categories. In *Proceeding of the 7th Annual Conference of the Cognitive Science Society*, pp. 283-287, Hillsdale, NJ, 1985.

Gordon A., *Classification*. 2nd edition. Chapman and Hall/CRC press, London, UK, 1999.

Halkidi M., Y. Batistakis, and M. Vazirgiannis, On clustering validation techniques. *Journal of Intelligent Information Systems* 17: 107-145, 2001.

Hargrove, J.L., & Schmidt F.H., The role of mRNA & protein stability in gene expression. *FASEB J*., **3**(12), 2360-70, 1989.

Hartemink, A.J., Gifford, D.K., Jaakola, T.S., & Young, R.A., Combining location & expression data for principled discovery of genetic regulatory network models. *Proc. Pac Symp Biocomput.*, **7**, 437-49, 2002.

Hartigan, J.A. & Wong, M.A., A k-means clustering algorithm. *Appl. Stat. J. Roy. St. C*., **28**, 100-108, 1979.

Holstege, F.C., Jennings, E.G., Wyrick, J.J., Lee, T.I., Hengartner, C.J., Green, M.R., Golub, T.R., Lander, E.S., & Young, R.A., Dissecting the regulatory circuitry of a eukaryotic genome. *Cell*, **95**(5), 717-28, 1998.

Holter, N.S., Maritan, A., Cieplak, M., Fedoroff, N.V., & Banavar, J.R., Dynamic modeling of gene expression data. *Proc. Natl. Acad. Sci. USA*., **98**(4), 1693-8, 2001.

Ideker, T., Thorsson, V., Ranish, J.A., Christmas, R., Buhler, J., Eng, J.K., Bumgarner, R., Goodlett, D.R., Aebersold, R., & Hood, L., Integrated genomic & proteomic analyses of a systematically perturbed metabolic network. *Science*, **292**(5518), 929-34, 2001.

Jain A.K., M.N. Murty, and P.J. Flynn, Data clustering: a review. *ACM Computing Surveys*, 31(3): 264-323, 1999.

Jakt, L.M., Cao, L., Cheah, K.S., & Smith, D.K., Assessing clusters & motifs from gene expression data. *Genome Res*., **11**(1), 112-23, 2001.

Jarvis, E.D., Smith, V.A., Wada, K., Rivas, M.V., McElroy, M., Smulders, T.V., Carninci, P., Hayashizaki, Y., Dietrich, F., Wu, X., McConnell, P., Yu, J., Wang, P.P., Hartemink, A.J., & Lin, S., A framework for integrating the songbird brain. *J. Comp. Physiol. A. Neuroethol. Sens. Neural Behav. Physiol*., **188**(11-12), 961-80, 2002.

Jeong, H., Mason, S.P., Barabasi, A.L. & Oltvai, Z.N., Lethality & centrality in protein networks. *Nature*, **411**(6833), 41-2, 2001.

Jin, J.Y., Almon, R.R., DuBois, D.C., Jusko, W.J., Modeling of corticosteroid pharmacogenomics in rat liver using gene microarrays. *J. Pharmacol. Exp. Ther*., **307**(1), 93-109, 2003.

K. Fukunaga, *Introduction to Statistical Pattern Recognition*, San Diego, CA: Academic Press, 1990.

Kamel M.S. and N. M. Wanas, Data dependence in combining classifiers. *Lecture Notes in Computer Science*, 2709: 1-14, 2003.

Karypis G. and V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359-392, 1998. Karypis G., R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: applications in VLSI domain. In *Proceedings of the Design and Automation Conference*, pp. 526-529, Anaheim, California, 1997.

Kel AE, Gossling E, Reuter I, Cheremushkin E, Kel-Margoulis OV, Wingender E., MATCH: A tool for searching transcription factor binding sites in DNA sequences. Nucleic Acids Res 31:3576-3579, 2003.

Kel, A.E., Gossling, E., Reuter, I., Cheremushkin, E., Kel-Margoulis, O.V., & Wingender, E., MATCH: a tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res*., **31**(13), 3576-9, 2003.

Kholodenko, B.N., Kiyatkin, A., Bruggeman, F.J., Sontag, E., Westerhoff, H.V., & Hoek, J.B., Untangling the wires: a strategy to trace functional interactions in signaling & gene networks. *Proc. Natl. Acad. Sci. USA*., **99**(20), 12841-6, 2002.

Kumar S.P., Feidler J.C., BioSPICE: A computational infrastructure for integrative biology. OMICS 7:225, 2003.

Kumar, R., Reynolds, D.M., Shevchenko, A., Shevchenko, A., Goldstone, S.D., & Dalton, S., Forkhead transcription factors, Fkh1p & Fkh2p, collaborate with Mcm1p to control transcription required for M-phase. *Curr. Biol*., **10**(15), 896-906, 2000.

Laerhoven K.V., Combining the self-organizing map and k-means clustering for on-line classification of sensor data. In *Proceedings of the International Conference on Artificial Neural Networks 2001* (ICANN'01), pp. 464-469, Vienna, 2001.

Lee, T.I., Rinaldi, N.J., Robert, F., Odom, D.T., Bar-Joseph, Z., Gerber, G.K., Hannett, N.M., Harbison, C.T., Thompson, C.M., Simon, I., Zeitlinger, J., Jennings, E.G., Murray, H.L., Gordon, D.B., Ren, B., Wyrick, J.J., Tagne, J.B., Volkert, T.L., Fraenkel, E., Gifford, D.K., & Young, R.A., Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, **298**(5594), 799-804, 2002.

Liao JC, Boscolo R, Yang YL, Tran LM, Sabatti C, Roychowdhury VP, Network component analysis: Reconstruction of regulatory signals in biological systems. Proc Natl Acad Sci USA 100:15522-15527, 2003.

Liu, X., Brutlag, D.L., & Liu, J.S., BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Proc. Pac. Symp. Biocomput.*, **4**, 127-38, 2001.

Matys, V., Fricke, E., Geffers, R., Gossling, E., Haubrock, M., Hehl, R., Hornischer, K., Karas, D., Kel, A.E., Kel-Margoulis, O.V., Kloos, D.U., Land, S., Lewicki-Potapov, B., Michael, H., Munch, R., Reuter, I., Rotert, S., Saxel, H., Scheer, M., Thiele, S., & Wingender, E., TRANSFAC: transcriptional regulation, from patterns to profiles. *Nucleic Acids Res*., **31**(1), 374-8, 2003.

Monti S., P. Tamayo, J. Mesirov, and T. Golub, Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52:91-118, 2003.

Nadon, R., & Shoemaker, J., Statistical issues with microarrays: processing and analysis, 2002.

Neves, S.R., & Iyengar, R., Modeling of signaling networks. *Bioessays*, **24**(12), 1110-7, 2002.

Niehrs, C., & Pollet, N., Synexpression groups in eukaryotes. *Nature*, **402**(6761), 483-7, 1999.

Niethammer, M.N., Menold, P.H., & Allgöwer, F., Parameter & derivative estimation for nonlinear continuous-time system identification. *Proceedings of the 5th IFAC symposium on Nonlinear systems*, 691-6, 2001.

Panda S, Antoch MP, Miller BH, Su AI, Schook AB, Straume M, Schultz PG, Kay SA, Takahashi JS, Hogenesch JB, Coordinated transcription of key pathways in the mouse by the circadian clock. Cell 109:307-320, 2002.

Patra, A., & Unbehauen, H., Identification of a class of nonlinear continuous time systems using Hartley modulating functions. *Int. J. Control*, **62**(6), 1431-51, 1995.

Pennisi, E., Human genome. A low number wins the GeneSweep Pool. *Science*, **300**(5625), 1484, 2003.

Quandt, K., Frech, K., Karas, H., Wingender, E., & Werner, T., MatInd & MatInspector: new fast & versatile tools for detection of consensus matches in nucleotide sequence data. *Nucleic Acids Res.*, **23**(23), 4878-84, 1995.

Ramakrishnan, R., DuBois, D.C., Almon, R.R., Pyszczynski, N.A., Jusko, W.J., Fifth-generation model for corticosteroid pharmacodynamics: application to steady-state receptor down-regulation and enzyme induction patterns during seven-day continuous infusion of methylprednisolone in rats. *J. Pharmacokinet. Pharmacodyn*., **29**(1), 1-24, 2002.

Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N. & Barabasi, A.L., Hierarchical organization of modularity in metabolic networks. *Science*, **297**(5586), 1551-5, 2002.

Roth, F.P., Hughes, J.D., Estep, P.W., & Church, G.M., Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.*, **16**(10), 939-45, 1998.

Schmitt, A.P., & McEntee, K., Msn2p, a zinc finger DNA-binding protein, is the transcriptional activator of the multistress response in *Saccharomyces cerevisiae. Proc. Natl. Acad. Sci. USA.*, **93**(12), 5777-82, 1996.

Sebastiani, P., Gussoni, E., Kohane, I.S., & Ramoni., M.F., Statistical Challenges in Functional Genomics. *Statist. Sci.*, **18**(1), 33-70, 2003.

Selinger. D.W., Saxena, R.M., Cheung, K.J., Church, G.M., & Rosenow, C., Global RNA half-life analysis in *Escherichia coli* reveals positional patterns of transcript degradation. *Genome Res.*, **13**(2), 216-23, 2003.

Sherlock, G., Analysis of large-scale gene expression data. *Curr. Opin. Immunol.*, **12**(2), 201-5, 2000.

Shinbrot, M., On the analysis of linear & nonlinear systems. *T. Am. Soc. Mech. Eng.*, **79**, 547-552, 1957.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., & Futcher, B., Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell.*, **9**(12), 3273-97, 1998.

Strehl A. and J. Ghosh, Cluster ensembles: a knowledge reuse framework for combining partitionings. *Journal of Machine Learning Research*, 3:583-617, 2002.

Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J. & Church, G.M., Systematic determination of genetic network architecture. *Nat. Genet.*, **22**, 281-5, 1999.

Taylor, I.A., McIntosh, P.B., Pala, P., Treiber, M.K., Howell, S., Lane, A.N., & Smerdon, S.J., Characterization of the DNA-binding domains from the yeast cell-cycle transcription factors Mbp1 & Swi4. *Biochemistry.*, **39**(14), 3943-54, 2000.

Tegner, J., Yeung, M.K., Hasty, J., & Collins, J.J., Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling. *Proc. Natl. Acad. Sci. USA.*, **100**(10), 5944-9, 2003.

Topchy A., A.K. Jain and W. Punch, A mixture model for clustering ensembles. In *Proceedings SIAM Conference on Data Mining*, pp. 379-390, 2004.

Topchy A., A.K. Jain and W. Punch, Combining multiple weak clustering. In *Proceedings IEEE International Conference on Data Mining*, pp. 331-338, Melbourne, Fl, 2003.

Trends Genet., **18**(5), 265-71.

Vadigepalli, R., Chakravarthula, P., Zak, D.E., Schwaber, J.S., & Gonye, G.E., PAINT: A promoter analysis & interaction network generation tool for genetic regulatory network identification. *Omics,* **7**(3), 235-252, 2003.

Van Someren, E.P. , Wessels, L.F.A, & Reinders, M.J.T., Linear modeling of genetic networks from experimental data. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **8**, 355-66, 2000.

Van Someren, E.P., Wessels, L.F.A, Reinders, M.J.T., & Backer, E., Searching for limited connectivity in genetic network models. *Proc. 2nd Intl. Conf. Systems Biology*, 222-30, 2001.

Wang, Y., Liu, C.L., Storey, J.D., Tibshirani, R.J., Herschlag, D. & Brown, P.O., Precision & functional specificity in mRNA decay. *Proc. Natl. Acad. Sci. USA*, **99**(9), 5860-5, 2002.

Weaver, D.C., Workman, C.T., & Stormo, G.D., Modeling Regulatory Networks with Weight Matrices. *Proc. Pac. Symp. Biocomput.*, **4**, 112-23, 1999.

Wilusz, C.J., Wormington, M., & Peltz, S.W., The cap-to-tail guide to mRNA turnover. *Nat. Rev. Mol. Cell. Biol.*, **2**(4), 237-46, 2001.

Yang M.S. and K.L. Wu, A similarity-based robust clustering method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4): 434-448, 2004.

Yang, E., Van Nimwegen, E., Zavolan, M., Rajewsky, N., Schroeder, M., Magnasco, M., & Darnell, J.E. Jr., Decay rates of human mRNAs: correlation with functional characteristics and sequence attributes. *Genome Res.*, **13**(8), 1863-72, 2003.

Yeung, M.K., Tegner, J., & Collins, J.J., Reverse engineering gene networks using singular value decomposition & robust regression. *Proc. Natl. Acad. Sci. USA.*, **99**(9), 6163-8, 2002.

Yokobayashi, Y., Collins, C.H., Leadbetter, J.R., Weiss, R., & Arnold, F.H., Evolutionary design of genetic circuits and cell-cell communications. Adv. Complex. Syst., **6**, 37-45, 2003.

Zak, D.E., Doyle, F.J. III, Gonye, G.E., & Schwaber, J.S., Simulation studies for the identification of genetic networks from cDNA array & regulatory activity data. *Proc. 2nd Intl. Conf. Systems Biology*, 231-8, 2001.

Zak, D.E., Gonye, G.E., Schwaber, J.S., & Doyle, F.J. III, Importance of input perturbations & stochastic gene expression in the reverse engineering of genetic regulatory networks. *Genome Res*., **13**(11), 2396-2405, 2003a.

Zak, D.E., Pearson, R.K., Vadigepalli, R., Gonye, G.E., Schwaber, J.S., & Doyle, F.J. III, Continuous-time identification of gene expression models. *Omics*, **7**(4), 373-386, 2003b.

Zhang X. Fern and C.E. Brodley, Random projection for high dimensional data clustering: a cluster ensemble approach. In *Proceedings of 20th International Conference on Machine Learning (ICML)*, pp. 186-193, Washington DC, 2003.

Zhang X. Fern and C.E. Brodley, Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of 21st International Conference on Machine Learning(ICML)*, Banff, Alberta, Canada, 2004.

Zhu, J., & Zhang, M.Q., SCPD: a promoter database of the yeast *Saccharomyces cerevisiae*. *Bioinformatics.*, **15**(7-8), 607-11, 1999.

11 List of Acronyms

Acronym	Description
CE	Composite Elements
ChIP	Chromatin Immunoprecipitation
CIM	Candidate Interaction Matrix
CRE	Cyclic-AMP Response Element
EGF	Epidermal Growth Factor
GUI	Graphical User Interface
IOD	Input-Output Differential
KAGAN	Karyote Genome Analyzer
MCB	*Mlu*I cell-cycle Box
MF	Modulating Function
NCA	Network Component Analysis
ORF	Open Reading Frame
OAA	Open Agent Architecture
PAINT	Promoter Analysis and Interaction Network Tool
SCB	Swi4-dependent cell-cycle Box
SCN	SupraChiasmastic Nucleus
SCPD	*S. cerevisae* Promoter Database
SEB	Staphylococcal Enterotoxin B
SSE	Sum of Squared Errors
TF	Transcription Factor
TRE	Transcriptional Regulatory Element
TRNA	Transcriptional Regulatory Network Analysis
TSS	Transcription Start Sites
UTR	Untranslated Region